Using Visual Website Similarity for Phishing Detection and Reporting

Max-Emanuel Maurer
University of Munich
Amalienstr. 17
80333 Munich, Germany
max.maurer@ifi.lmu.de

Dennis Herzner
University of Munich
Amalienstr. 17
80333 Munich, Germany
herzner@cip.ifi.lmu.de

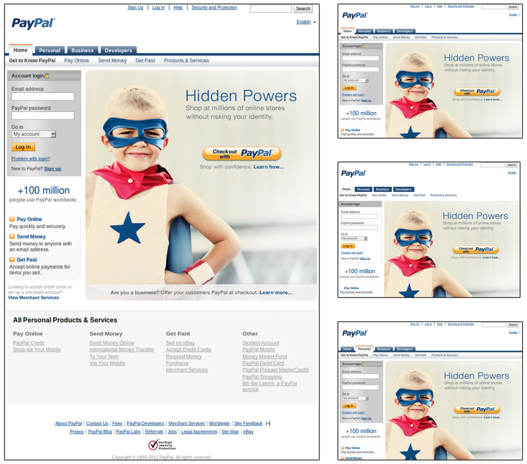

PayPal
Hidden Powers
Shop at millions of online stores without risking your identity.
+100 million
All Personal Products & Services

Figure 1: Legitimate paypal.com website (left) and three phishing attacks (right) taken from a test set.

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract

Phishing is a severe threat to online users, especially since attackers improve in impersonating other websites [1]. With websites looking visually the same, users are fooled more easily (see figure 1). However, the close visual similarity can also be used to counteract phishing. We present a framework that uses visual website similarity: (1) to detect possible phishing websites and (2) to create better warnings for such attacks. We report first results together with the three step process planned for the project. We expect the detection results to be comparable to previously published work which would allow for new kinds of phishing warnings with better coverage, less false positives and explicit user recommendations how to avoid these critical situation.

Keywords

Phishing Detection, Image Comparison

ACM Classification Keywords

K.4.4 [**Electronic Commerce**]: Security

General Terms

Security

Introduction

Phishing is a severe threat to online users. A major online community dedicated to finding phishing websites, has identified over 160,000 attacks in 2011[1]. How many attacks stay undetected is unknown. Most of these try to impersonate famous brands like Facebook, HSBC or Paypal to get user credentials [11].

To protect users, it is most common to identify and validate the existing phishing websites using different means and to block access by using blacklists [8]. But adding websites to blacklists takes time during which users are vulnerable to attacks.

Users are best fooled by websites looking exactly like the originals [6]. In these cases only security indicators outside of the main browser window remain to detect an attack. But those passive indicators are usually overlooked [7].

We use this perfect visual similarity of websites for counteracting the attacks. By comparing a website that is currently rendered on the users computer against other legitimate images, attacks can be identified. Huge databases of website images are already available[2].

We present our framework for detecting phishing websites through visual comparison. We provide a comparison backend server and an exemplary browser plugin that is able to query our backend system. Since image comparison can only provide a probability score for an attack, design and evaluation of a GUI will be an important last step.

Related Work

Visual comparison for the detection of phishing websites has been proposed several times. Mostly in conjunction with other heuristics for phishing detection.

Wenyin et al. [13] presented a concept that uses three types of similarity to detected phishing websites: ‘block level’, ‘layout’ and ‘overall style similarity’. Medvet et al. [10] propose a system that computes a website signature out of three features of a web page: visible text sections, embedded images and the overall visual appearance. Signatures can then be compared to other signatures. They evaluated the detector against a set of 140 phishing websites and 27 real websites performing very well. Chen et al. [4] use the rendered web page as input to a normalized compression distance compressor. With a test set of 320 phishing websites that target 16 different banking websites they showed that phishing websites are rated significantly closer to their originals than banking pages among themselves.

The prior work shows that detecting phishing websites through visual similarity works well in general. With our work we further elaborate the idea by finding an optimal detector, developing a client-server based detection framework and by investigating user interfaces for such a concept.

The Concept

Our main goal is to use visual similarity between real and fake websites for two applications: (1) to automatically detect websites that visibly impersonate others and (2) use the similarity between the images together with the difference in website URLs to make the user aware of the possible fake website whilst providing a proper alternative.

[1] www.phishtank.com

[2] e.g. www.searchpreview.de

We carry this out using a three step process. (1) First we build a detector to find out which (out of five) image comparison methods is best suited for the detection of similar phishing websites. To test every method we use a large set (> 1000) of known phishing websites and their legitimate counterparts. (2) Secondly the infrastructure for a web based detector and a way to query the detector is set up. (3)In the end we create and evaluate a user front-end that guides the user whenever a possible phishing attack is detected. Currently most of the first and second step has already been realized. For the user interface first drafts exist. Both will be presented in the remainder of this paper.

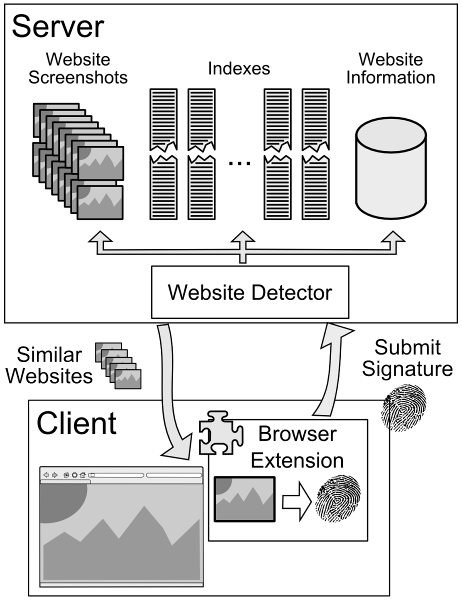


Figure 2: Architecture of the system. On the server side the detector analyzes fingerprints sent from the client and returns similar websites and additional information.

The Comparison Framework
The comparison framework is built as a client server architecture (see figure 2). The server stores a database of URLs, additional information and screenshots of different websites. Additional index-files contain the signatures of the screenshots depending on the detector used. The server can take an image or a precomputed signature as input for comparison inside the website detector.

For comparison the client precomputes a signature and sends it to the server. Images are not sent for speed and privacy reasons. To guarantee that the locally rendered image is comparable with the images stored in the image database, rendering of the website needs to be consistent for different clients – e.g. using the same resolution.

Given a signature of a potential phishing website, it can either be compared against a list of legitimate websites to find the website that is impersonated or against a list of known phishing websites to find fraudulent websites on other domains. The second approach makes sense since phishers usually deploy the same phishing website on a large number of different URLs.

Website Detector
The Java library LIRe [9] is used to extract signatures from the rendered representations of the websites in the database and stores them in a lucene[3] index. For performance reasons every feature type has its own index and a stored document only contains data for a single feature type together with a key to find the corresponding website in the database.

A search returns similar images together with a similarity score. These images are combined with URL information

[3] lucene.apache.org

from the database. High similarity will also exist e.g. between the legitimate website and its own screenshot on the server. After separating those, a list of similar domains together with the similarity score and URL information is passed back to the client.

Detectors Used
To find the best feature type we test five different descriptors: Scalable Color (SCD) and Color Layout (CLD) are descriptors proposed by the MPEG-7 standard [5]. SCD is a color histogram in the HSV color space that characterizes the global color distribution in the image. By encoding the histogram using a haar transform and discarding coefficients its size is reduced. CLD describes the spatial distribution of colors in the YCbCr color space by subdividing the image into 64 blocks and calculating a representative color (e.g. the average color) for each block.

More sophisticated Compact Composite Descriptors CEDD, FCTH and JCD can combine more than one feature in a single histogram [3]. The Color Edge Directivity Descriptor (CEDD) and Fuzzy Color and Texture Histogram (FCTH) include color and texture information in a very compact form, which makes them preferable to use in large datasets. Both extract color information by using two fuzzy systems to map the colors in a 24-color palette. For the texture information CEDD uses the filters proposed by the MPEG-7 Edge Histogram Descriptor. For FCTH a fuzzy system creates an 8-bin histogram that uses the high frequency bands of a haar wavelet transform. The Joint Composite Descriptor (JCD) is the combination of CEDD and FCTH [2]. Since both use the same color information JCD only maps their texture information into seven texture areas.

Evaluation
For the evaluation of the system we use a set of at least 1,000 phishing websites and a second set of legitimate website that have been impersonated by those websites mixed with the 1,000 most visited websites worldwide[4]. In a first step we will define the best detection thresholds using a subsample of the given websites.

Planned Tests
Our first goal is to find out which descriptor is best suited for the task of finding similar webpages. Therefore different websites will be evaluated against both sets. Any given phishing website will be checked to find out (1) how many legitimate websites can be detected and (2) whether similar phishing websites really target the same brand. For our set of legitimate websites it is important to test that they do not trigger similarity alerts against other legitimate websites. The numbers on false positives and false negatives will show which detector is best suitable for the task and how well our framework compares to the findings of related work. Besides accuracy, speed is another important measure.

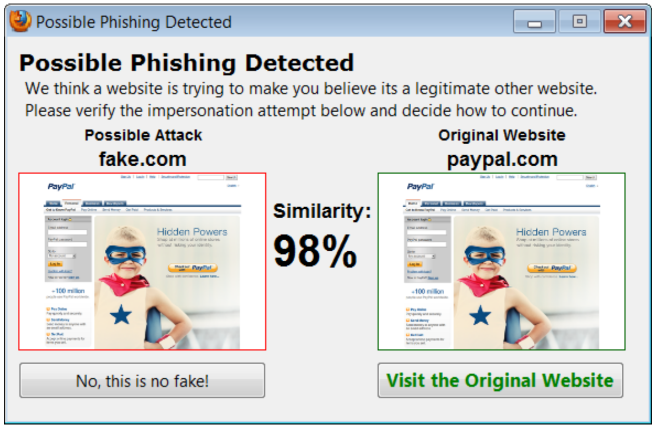
Expected Outcomes
We expect that our evaluation results will be able to compete with the results that have been achieved in related work. Our first tests showed good accuracy and speed using a smaller dataset. Searching an index of 2,100 websites took around 30ms. Using the fast lucene index guarantees a linear runtime $O(n)$. For huge datasets this might be a problem requiring additional work.

Challenges

There are several special kinds of websites that are challenging: Some websites contain large images – e.g.

[4] derived from alexa.com

commercials – as part of the layout that change with every reload of the webpages. Those large image sections could eventually influence the detection process negatively. Storing multiple versions of a website might help.

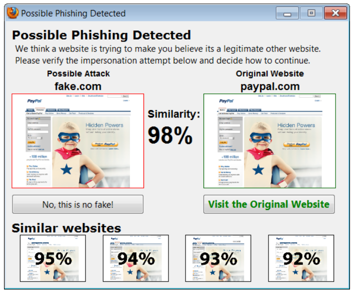


We think a website is trying to make you believe its a legitimate other website. Please verify the impersonation attempt below and decide how to continue.

Figure 3: First dialog draft of a possible user interface dialog.

Animations and other time-dependent multimedia content on websites are also a problem. A website would look different depending on when the screenshot was taken since the start of the multimedia content. Taking multiple screenshots at fixed times could help here. Login-websites – as usually targeted by phishers – have those characteristics more seldomly.

Another major problem is the redesign of a websites or brand as it happens from time to time. The previously indexed images would than be invalid. To solve this, reindexing has to occur after a while. This can be detected through client submissions with very different signatures for a known URL.



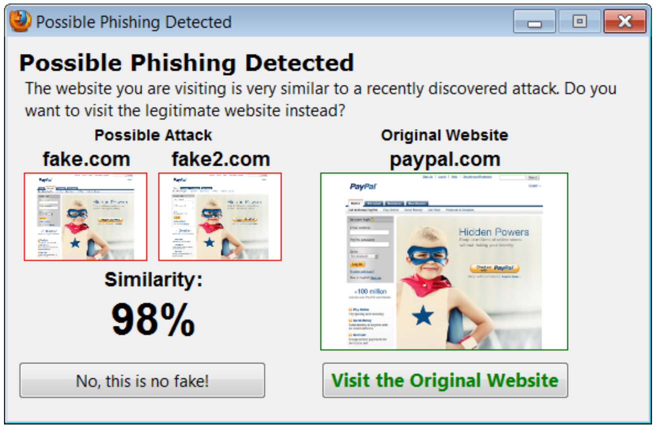
We think a website is trying to make you believe its a legitimate other website. Please verify the impersonation attempt below and decide how to continue.

Figure 4: A dialog draft containing multiple similar images that have been found.

User Interface Design

We expect that visual comparison of the websites will lead to a high accuracy in detection of possible phishing websites. This reduces unnecessary warnings and thus habituation.

In case a match has been found the warning should be actively interrupting the user from further browsing on this website. She can then review the detected similarity and decide how to proceed. Providing the user with possibilities on how to continue is important [12]. Since our approach already has a clue which legitimate website has been attacked the user can be given the option to navigate to this website instead.



The website you are visiting is very similar to a recently discovered attack. Do you want to visit the legitimate website instead?

Figure 5: A dialog for a detected phishing match.

Different drafts of the final interface can be found in figure 3 and 4. To make it easier for the user to understand why the warning has appeared and what the options are, we reduce the content of the warning as much as possible. The different URLs and similar images are intended to be catching content for the user, to verify the warning and to decide on the correct option. Continuing to the suspected phishing website should be less prominent than to the legitimate match. In case a similarity of a potential phishing site to another phishing site has been detected, it is also important to display this triangular relationship. One example draft for this can be found in figure 5. All those drafts have not been implemented or evaluated yet. Before implementing the user interface as a prototype, we will discuss possible designs in a focus group.

Evaluation
We plan to roll out the final prototype as a browser plugin to the public for a field study. We will measure data on appearing dialogs and decisions users take. Different questionnaires are planned throughout the study to capture user experience.

Conclusions

In this paper we presented our three step concept for enhancing automatic phishing detection through visual image comparison of rendered websites. Related work shows that this can help to identify phishing attacks before they are manually listed on a blacklist. In the current state of this project, different image detectors are compared to find an optimal detector for our use case. A client-server architecture and user interface drafts have also been created and shall be evaluated as a next step.

Acknowledgements

We thank Doris Hausen for her help during the final steps of this paper.

References

[1] M. Blythe, H. Petrie, and J. A. Clark. F for fake: Four studies on how we fall for phish. In *CHI*, 2011.

[2] S. A. Chatzichristofis, A. Arampatzi, and Y. S. Boutalis. Investigating the behavior of compact composite descriptors in early fusion, late fusion and distributed image retrieval. *Radioengineering*, 2010.

[3] S. A. Chatzichristofis, K. Zagoris, Y. S. Boutalis, and N. Papamarkos. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *IJPRAI*, 2010.

[4] T. Chen, S. Dick, and J. Miller. Detecting visually similar web pages. *Transactions on Internet Technology*, 2010.

[5] L. Cieplinski. Mpeg-7 color descriptors and their applications. In *CAIP*, 2001.

[6] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *CHI*, 2006.

[7] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *CHI*, 2008.

[8] Google Inc. Google safe browsing. www.google.com/tools/firefox/safebrowsing [Last visited: 2012, Jan 4th].

[9] M. Lux and S. A. Chatzichristofis. Lire: lucene image retrieval: an extensible java cbir library. In *MM*, 2008.

[10] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In *SecureComm*, 2008.

[11] OpenDNS. Web content filtering and phishing. *OpenDNS 2010 Report*, 2010.

[12] T. Roessler and A. Saldhana. W3C: Web security context: User interface guidelines. www.w3.org/TR/wsc-ui/ [Last visited: 2012, Jan 4th], 2010.

[13] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng. Detection of phishing webpages based on visual similarity. In *WWW*, 2005.