Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity Against Phishing

Max-Emanuel Maurer[1] and Lukas Höfer[1]

University of Munich
Media Informatics Group
Amalienstr. 17
803333 Munich
Germany
`max.maurer@ifi.lmu.de`, `hoeferl@cip.ifi.lmu.de`

Abstract. Phishing attacks rise in quantity and quality. With short online lifetimes of those attacks, classical blacklist based approaches are not sufficient to protect online users. While attackers manage to achieve high similarity between original and fraudulent websites, this fact can also be used for attack detection. In many cases attackers try to make the Internet address (URL) from a website look similar to the original. In this work, we present a way of using the URL itself for automated detection of phishing websites by extracting and verifying different terms of a URL using search engine spelling recommendation.
We evaluate our concept against a large test set of 8730 real phishing URLs. In addition, we collected scores for the visual quality of a subset of those attacks to be able to compare the performance of our tests for different attack qualities. Results suggest that our heuristics are able to mark 54.3% of the malicious URLs as suspicious. With increasing visual quality of the phishing websites, the number of URL characteristics that allow a detection increases, as well.

1 Introduction

Phishing as the act of stealing personal data of Internet users for misuse is an old but still threatening problem. As the number of Internet users and online transactions grows, the possibility of misuse is also growing. 164,917 websites targeting users to input sensitive information – like passwords – have been recorded by the online service phishtank.com in 2011 [1]. These are only the phishing websites that have been reported and detected by a single community. Phishing is hence an important cyber security problem. Google recently announced that 9,500 websites are added to their blacklist each day [2]. Projecting this to a whole year more than three million URLs are recorded each year. Nowadays, phishers use sophisticated software toolkits to launch a large number of Phishing websites on different URLs to counteract common security methods like blacklists [3] that are most commonly used as phishing protection. With additional backdoors in those phishing toolkits, phishers even target each other [4].

Since it takes some time to get websites on blacklist indexes, first visitors to phishing websites are left vulnerable [5]. Using many different URLs [6] in combination with intelligent botnets that hide their master-servers using the fast-flux methods [7] it gets nearly impossible to block those attacks using conventional methods.

Users could spot most of the phishing attacks by themselves by closely examining URLs and other indicators with the right amount of security knowledge and focus, but as security is never the users' primary goal [8] they fail to detect most of the attacks.

Another issue is that phishers usually try to closely impersonate a trusted party the user knows by imitating brands, website design, logos or as a special case the URLs. This being an issue for users to fall for phishing [9], it should be used as an input to online security research to generate new means for phishing detection.

In this work we want to focus on URL similarity. Since phishers usually can not use the exact same URL they are targeting, they use different deceptions to build domain names and paths that look similar to the original domain. As an example they use small spelling mistakes that might be overlooked by the user. In case the spelling of a phishing URL is close to a real domain name or brand name automatic detection of phishing attacks becomes possible. We present such a detection approach using URL terms together with search engine spelling suggestions.

To test this we gathered a data set of 8730 phishing websites and looked at the different kinds of attacks that can be found and how they could be detected through similarity matching. For our tests we simply used queries sent to search engines that support spelling corrections for the submitted query. In our case we used terms extracted from the URL as a query to the search engine.

In our opinion the visual quality of the attacks also plays an important role and we were interested to find out how the perceived visual quality of a phishing attack correlates with our detection results. For a subset of our test websites we had experts rate the visual deception quality for each single website. We defined that as how identical the phishing website itself looks to its original or rather how well it is designed. Using those ratings we were able to infer whether our detection methods hold better for high or low quality phishing websites.

2 Related Work

To be able to understand phishing and find new methods to protect users from sophisticated attacks it is important to look at how people fall for phishing attacks and how attackers design their attacks to make them “appealing” for their victims. Human Computer Interaction or more specific the field of Usable Security addresses security questions from a user perspective.

Research on phishing in HCI is bound to different domains. Understanding the problem, detecting attacks and finally how to communicate the detection results to the user for a final decision.

When trying to detect malicious websites many software toolkits or browser toolbars fail. In 2006 Wu et al. [10] tested three different browser toolbars and found them all being ineffective in preventing phishing attacks. Numerous approaches to detect phishing websites without the need of manual verification have been presented since then. Besides using the source code of a website or the visual similarity between the rendered content of other websites, two recent publications also partially take URLs into account for their detection process.

SpoofGuard by Chou et al. [11] uses the domain name, URL, links and images to compute the likeliness of an attack. In case certain patterns appear in the URL (e.g. using the @-symbol or IP-addresses) the probability score for an attack is increased. Together with a variety of other tests (e.g. non-ssl forms with password fields) a total spoof score (TSS) is computed.

Zhang et al. [12] presented a similar tool called CANTINA in 2007. As the most important part they use the TF-IDF-algorithm to find the most typical written terms in a web page. As these terms should be unique to the given website a search engine query should point back to the website from where the terms originally were gathered. This concept of so called “robust hyperlinks” has been presented earlier by Phelps and Wilensky [13]. For their evaluation Zhang et al. used 100 phishing pages and 100 legitimate pages and submitted the respective terms to a search engine. They also tried to add the domain name to the found terms which reduced the number of false but also of true positives. In a second experiment they added additional heuristics (similar to SpoofGuard) to their computation. In 2011 Cantina was enhanced as “Cantina+” [14] to a machine-learning based system that uses 15 different features. In their approach six of those features are URL based features. When testing the new approach over 92% of true positives for phishing websites in the test set were detected.

In all publications the URL only plays a minor role for the detection of potential phishes and if taken into account only specific characteristics are used. For our work we focussed on the URLs to find the the potential of phishing detection that comes out of the sole use of the URL and its subterms. Although we focus on this specific topic, sophisticated detection should always use more than one way.

3 URL phishing detection

When hosting phishing websites phishers usually try to pick URLs that look trustful or well known to the user. Being able to create a perfect visual copy of a website, the domain name remains as one of the last resorts to detect that the website, the user is currently visiting, denotes an attack. Many of these attacks are well known and have been collected and reported in related work [15]. The homograph attack is one sophisticated example for these spoofing attacks where similar looking characters form other languages are used to register internationalized domain names that look (nearly) perfectly the same on the screen as their original ones [16].

To detect those misspellings simple algorithms like the Levenshtein distance [17] could be used. It counts the minimum numbers of insertions, deletions and substitutions that are required to transform one term into another. “paypai.com” (please notice the ‘i’ instead of the ‘l’) and “paypal.com” would have a Levensthein distance of 1. Website URLs with a small distance to a similar legitimate URL hence would be suspicious. To perform those comparisons a large list of legitimate URLs and brand names would be needed.

In our approach we chose to use an existing environment to look for similar domain names or hidden brand names in the URL by using the spell checking functionality of a search engine. Sending a similar domain name to a search engine usually returns a suggestion to the search results for the more prominent name of the original website. For example if "paypai.com" is sent to a search engine it will return a suggestion to search for “paypal.com” instead, as this search term and its results are much more prominent. This knowledge about a vast number URLs together with their importance is what we make use of for attack detection.

3.1 Possible Subterms

Using the whole URL including all path information would not yield any valuable results. After conducting some first trials, we found that for the mostly long queries, small spelling mistakes do not return usable suggestions. Hence we developed algorithms to detect possible search terms that are worth checking. We derived those from common attacks in literature and from what we found during the analysis of existing phishing URLs. Table 1 shows four example URLs with the extracted terms highlighted. The four cases are as follows:

- **Basename:** The base name is the real domain name as registered at the registrar for the domain. The basename usually consists of the top-level domain (e.g. ‘com’) and the domain name itself (e.g. ‘paypal’). Phishers cannot use the original domain name as it is already registered by the original company. Instead, they register misspellings or similar looking domain names.
- **Subdomains:** For each base domain, the owner can specify an arbitrary number of subdomains. This is often used to prepend the domain name of the websites that is attacked. Prepending the subdomains “us.battle.net” to any other domain may fool users into thinking that they are on the real domain. Domain highlighting in the browser’s location bar is used by browser vendors to counteract such attacks but users are still being tricked by them [18].
- **Pathdomain:** In some cases, phishers neither have access to the base name or a subdomain (e.g. when hosting their attack on a free web hosting service). In this case, they place a second domain as a subfolder of the URL path – usually right after the domain name. For the remainder of this paper, will refer to those terms as “pathdomains”.
- **Brand name:** A last check we performed, was for certain brand names. In some cases not a whole domain but only a brand name is inserted somewhere in the URL. For this special case we did not use the help of a search engine. Instead, we only counted the sole occurrence.

Type	Sample-URL
Basename:	http://www.warldofworcarft.com
	Attackers picked a domain with some typos. "warldofworcarft" might be easily misread as "worldofwarcraft".
Subdomain:	http://us.battle.net.loginaccountbattle.net/login/en/login.html
	A misleading subdomain is preceeding the real domain that hosts the attack. "us.battle.net" usually already is the full domain name. In this case it just serves as arbitrary subdomains to the real domain "loginaccountbattle.net".
Pathdomain:	http://piasel.altervista.org/www.paypal.com/new/paypal/intl/update/
	The domain is created as a folder on the upmost level of the server. The attack claims to be "www.paypal.com" which is put next to the real domain name "altervista.org".
Brand Name:	http://www.radiotelemiracle.com/includes/Archive/NATWEST/index.html
	In this case no full domain name is included but only the brand name is part of the URLs path argument. For this work we used a simple list of 21 brands.

Table 1. Examples of the four different URL patterns that were extracted from the URLs.

3.2 URL Extraction

The algorithms we use for extracting the terms are quite simple and dependent from each other.

The base domain can be extracted by firstly finding a valid public TLD (top-level domain) suffix at the end of the whole domain name (without any path information). Top-level domains denote the highest level in the Internet domain name system (DNS) [19]. In most cases the last characters after the last dot form the top-level domain name. In some countries a secondary level is added before customers can register their own arbitrary names (e.g. "co.uk"). For "www.paypai.com" the TLD would be ".com". To complete the base domain the next preceding domain part ("paypai") is appended ("paypai.com"). Usually the resulting base domain has only two components. In case the registrar uses more than one level for the TLD the number of levels can increase (e.g. "payppai.co.uk"). Using a rule set together with a list of such special cases all base domains can be easily found [20].

Knowing the base domain the remaining prepending domain parts construct the subdomains of the URL. By subtracting "paypai.com" from "www.paypai.com" only "www" remains as a subdomain in this case.

To look for pathdomains the process of finding a basedomain is repeated for the path portion of the URL – everything behind the first slash in the URL. If a valid TLD can be found anywhere in the path another domain part is prepended if applicable.

For our brand name validation we simply use a text search on the URL strings to find any brand name on our list. Since we just wanted to do some baseline testing we only included 21 brand names of brands that are attacked most often (see table 2).

After having identified the different domain parts we send them to a search engine and check the search engine results for a spell checking suggestion. In

53.com	Chase	Microsoft
ANZ	Citibank	Paypal
AOL	eBay	USBank
Banamex	E-Gold	Visa
Bankofamerica	Google	Warcraft
Barclays	HSBC	Westpac
battle.net	Lloyds	Yahoo

Table 2. The 21 brand names used for the brand name testing.

case a suggestion occurs the handed in query is most likely to be a misspelled domain and is hence counted as suspicious.

A system using this approach would be possibly deployed right at the user's web browser. The browser would be able to compute the different subdomains parts easily and could verify them by sending them to search engines or a specific server instance implementing our concept. We did not create any software component for end users as a part of this paper, as we were only interested in first measurements of the potential power of our concept.

4 Evaluation

For our evaluation we wanted to test our algorithms against a large test set of real phishing attacks and find out for how many of those URLs at least one of the four types of extractable cues exists. As a second evaluation we wanted to find out whether the number of suspicious websites we can find is dependent from the visual quality of the attack. Therefore a manual quality rating for each of our test websites was needed. Due to the immense manual workload, we reduced the number of test pages for this second step.

4.1 Methodology

The evaluation took two steps: Firstly we created a large set of URLs (8730 pages) gathered from phishtank to apply our four algorithms against them to extract possible search queries. We then submitted the queries to a major search engine and tested for each different entry whether any spelling suggestions would be returned and counted these. In a second phase we reduced our test set to 566 websites that we had captured with screenshots and had them rated by three expert Internet users (see section 4.3). For those websites, we looked at their test results again, this time incorporating the average quality rating that had been given by our experts. For the quality rating we also captured 127 original websites, and as a side effect, we could look on how our algorithms performed against those.

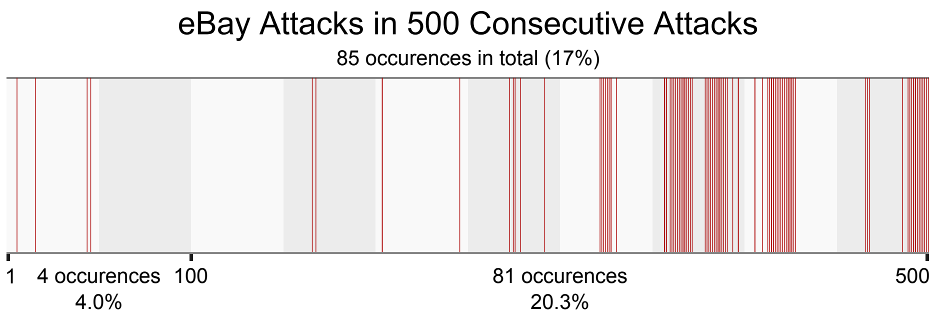


Fig. 1. Distribution of eBay attacks over the first 500 queried websites. The attacks are not equally distributed.

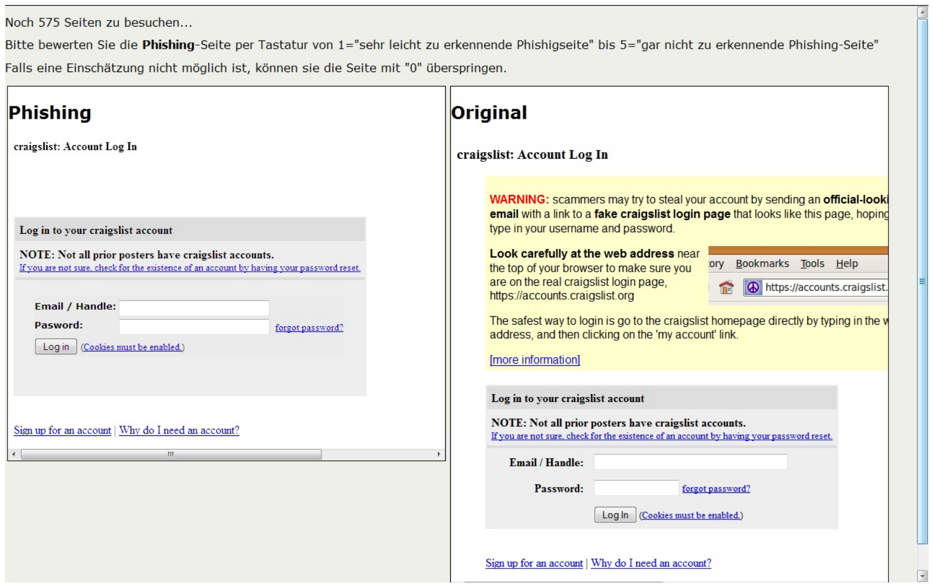
4.2 Building the Test Set

To acquire the test set, we used the online phishing website database phishtank.com[1]. We acquired websites from the phishing index for a period of about one month. Using this method we gathered 8730 different confirmed phishing URLs.

Particularly when testing tools to detect phishing attacks, a large test set is vital. On phisthank.com each attack is stored by its URL – which users received for example through an email. Many attacks of the same kind are often launched together sending out different URLs to different receipients. For a small sample set the amount of overall attacks of a certain kind may look totally different. When we compared our testing results of the first 100 entries in our list to the first 500 entries we noticed big differences in algorithm performance. A potential reason for that can be seen in figure 1. We analyzed the attacks targeting eBay to see whether they are equally distributed. Looking at the position of all eBay attacks amongst the first 500 websites one can quickly see huge differences. We only had four websites attacking eBay under the first 100 of our URLs. Having only this data one could conclude that 4% of all attacks are eBay attacks. Looking at the remaining 400 attacks the amount of eBay attacks would seem to be about 20%. This clearly shows that a large test set is very important in our case.

In addition to the simple URL collection we needed screenshots of the pages to be able to rate their visual quality later on in the second phase of our evaluation. We rendered 566 screenshots for web pages throughout the 8730 URLs. Where possible we also tried to find the parent website to the phishing attack. This resulted in 127 additional non-phishing web pages with one often being parent of multiple attacks. Due to short lifetime of detected phishing attacks the websites we used for rendering our screenshots are scattered over the complete range of the 8730 URLs.

[1] phishtank.com is an online service collecting and validating phishing attacks with community members



Noch 575 Seiten zu besuchen...

Bitte bewerten Sie die **Phishing**-Seite per Tastatur von 1="sehr leicht zu erkennende Phishigseite" bis 5="gar nicht zu erkennende Phishing-Seite"

Falls eine Einschätzung nicht möglich ist, können sie die Seite mit "0" überspringen.

Phishing

craigslist: Account Log In

Log in to your craigslist account

NOTE: Not all prior posters have craigslist accounts.
If you are not sure, check for the existence of an account by having your password reset.

Email / Handle:
Pasword: forgot password?
Log in (Cookies must be enabled.)

Sign up for an account | Why do I need an account?

Original

craigslist: Account Log In

WARNING: scammers may try to steal your account by sending an **official-looki** **email** with a link to a **fake craigslist login page** that looks like this page, hoping type in your username and password.

Look carefully at the web address near the top of your browser to make sure you are on the real craigslist login page, https://accounts.craigslist.org

The safest way to login is go to the craigslist homepage directly by typing in the w address, and then clicking on the 'my account' link.

[more information]

Log in to your craigslist account

NOTE: Not all prior posters have craigslist accounts.
If you are not sure, check for the existence of an account by having your password reset.

Email / Handle:
Password: forgot password?
Log In (Cookies must be enabled.)

Sign up for an account | Why do I need an account?

Fig. 2. A screenshot of the web interface our experts used to rate the phishing attacks. The interface language chosen to suit our experts' mother tongue.

4.3 Rating of Websites

For rating of the websites we asked three expert Internet users – one IT consultant, one informatics students and one media informatics student – to rate the quality of the attacks on a five point Likert scale from "1-the attack can be discovered easily" to "5-it is hard to discover the attack". We wanted them to compare the visual deception quality of the attacks. For this reason we showed them the screenshot of the phishing attack and the original website side by side in an online interface (see figure 2). They were displayed to the experts in random order and the experts rated them using the number keys – 1 to 5 – on their keyboard. Using the '0' key they were able to skip an entry for later processing.

The screenshots only showed an image of the content of the website. The browser frame with other information like the URL or other security indicators was not present. Using this interface the experts could rate all phishing websites. Deliberately we did not instruct our experts in any way which characteristics they should use for their assessment.

5 Results

The results for our evaluation can be divided into three parts. Firstly, the results gathered from the whole set of 8730 phishing websites As a comparison to that we also had a look at the results for the smaller subset of 566 websites where we actually extracted the screenshots and the corresponding original websites.

In the end we had a look at what can found when combining those results with the different quality ratings taken form our exports.

5.1 The Whole Test Set

We were able to extract a basename for all but 265 of our 8730 phishing URLs. These URLs were just IP-addresses and thus had no basename. Spell checking returned a result for 961 of the remaining 8465 websites (11.4%). As with the basenames, subdomains could also only be queried for non-IP-URLs. 2119 returned a spell check result (25.0%).

Looking for a pathdomain in the remainder of the path of the URL, we were able to find 1522 second domains using our algorithm (17.4%). Sending those extracted domains to the search engine only 232 (2.7%) returned a spell checking result. In this case it is possible that the spell checking did not return any results because the domain names were already written correctly – please refer to the discussion section (see section 6) for more details on that.

Results	All Attacks		Rated Attacks		Non-Phishing	
	N=8730	%*	N=566	%	N=127	%
Basename Spelling Results	961	11,4	41	7,2	0	0,0
Subdomain Spelling Results	2119	25,0	144	25,4	0	0,0
Pathdomains Extracted	1522	17,4	43	7,6	22	17,3
Pathdomain Spelling Results	232	2,7	3	0,5	0	0,0
Brand Name Hits	2021	23,2	63	11,1	31	24,4

*Where applicable IP-Address-Domains where excluded for basename and subdomain percentages. In those cases N = 8465.

Table 3. Results for general URL extraction and search engine queries throughout the different conditions.

Finally, we ran our brand checker against the URLs and tested for 21 brand names. 2021 URLs contained at least one of the brand names (23.2%). Table 3 contains an overview on those values.

We also looked at how many pages would have at least triggered one feature. When looking at all spell checking results and the results from our brand name detector combined 4742 attacks (54.3%) were marked as suspicious, triggering at least one feature. Including all URLs that contained a path domain – instead of just using path domains that returned a spelling result – we would even get a coverage of 4958 attacks (56.8%).

Figure 3 shows a detailed diagram of exactly where which feature was triggered. The x-axis denotes the 8730 different websites that we tested and a colored bar indicates that the specific feature marked the website as being suspicious. Beneath those results for the single features the diagram shows all features in a cumulated way – for each bar at least one feature would have been triggered.

Looking at the number of features that were triggered 3195 websites triggered only a single feature (36.6%), 1319 websites triggered two features (15.1%), 246 websites triggered four features (2.8%) and only 13 websites triggered all possible four features (0.15%). In average a website that was marked as suspicious hence had 1.39 overlapping features confirming this (SD 0.6).

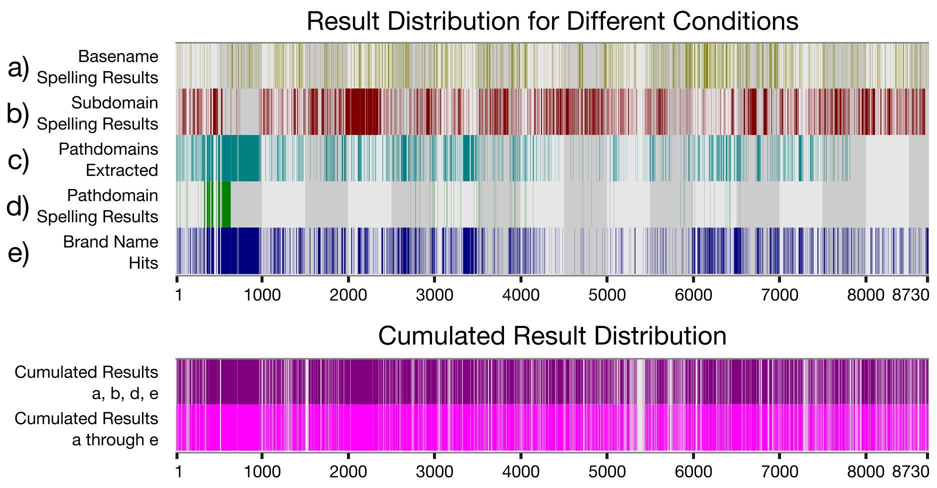


Fig. 3. Heatmaps showing the detailed matching results for the different domains in the different conditions. The lower part of the figures show all websites that triggered at least one search result.

5.2 Performance of the Subset

Looking at the same results for our subset of the 566 websites that had been captured for advanced testing the percentages are a little lower but the tendency is basically the same (see table 3). The main differences lie in the number of pathdomains that were extracted (7.6% instead of 17.4%) and their spelling results (0.5% instead of 2.3%). Another difference is in the number of brand hits (11.1% instead of 23.2%).

When running the tests on the 127 non-phishing websites that were captured spell checking did not find any suggestions for basename, subdomain and the pathdomains. The brand detector did detect 31 brands (24.4%) in the URL list. We expected those results for non-phishing URLs as they should not trigger any suggestions and certainly contained some brand names they represented.

5.3 Findings from the Quality Ratings

Looking at the rating results of our experts, we calculated an average score ranging from 1 to 5 for every phishing site's quality. This was the average of all three ratings our experts had given. According to those average ratings we partitioned our results into four intervals. “Very bad quality” [1-2], “bad quality”]2-3], “good quality”]3-4] and “very good quality”]4-5] by always including the next higher value into one interval excluding it from the next interval. Most of the attacks (226/39.9%) were rated to have a poor quality. The number of websites in each category constantly decreases to 79 (14.0%) with the highest ratings (see table 4).

Having distributed the attacks in those four categories we were able to recompute the different values for each category. Doing this it becomes clear that with rising quality of the phishing website the number of matches for our algorithms rises, too. For example the number of results for the basename check increases from 7.5% for very bad quality websites to 20.3% for very good quality websites (see table 4). The high number of poorly rated websites could hence also account for the overall lower detection values in our subset.

Quality	very bad [1;2]		bad]2;3]		good [3;4]		very good]4;5]	
	N=226	%	N=149	%	N=112	%	N=79	%
Basename Spelling Results	17	7,5	16	10,7	12	10,7	16	20,3
Subdomain Spelling Results	83	36,7	26	17,4	22	19,6	11	13,9
Pathdomains Extracted	9	4,0	11	7,4	13	11,6	10	12,7
Pathdomain Spelling Results	0	0,0	1	0,7	2	1,8	0	0,0
Brand Name Hits	18	8,0	16	10,7	16	14,3	13	16,5

Table 4. Results for the 566 visually rated URLs split up by rating intervals. Hit rates increase with quality.

6 Discussion and Limitations

Our evaluation yielded a lot of interesting results. However, looking at the overall number of websites that triggered spell checking results, this kind of detection mechanism will not be suitable as a sole detection for phishing websites. Hence, we suggest to use this method of spell checking URLs in combination with other methods. Our algorithms could be used for example, as an enhancement to existing score-based detectors.

Additionally, the results of this study only report on situations where a spell checking mechanism of a search engine returned a suggestion. We did not manually verify those suggestions for what causes they had (e.g. a homograph attack). In many cases, this might be equal to finding a possible phishing attack, but there might also be situations where the suggestions of the search engine had other causes. The tests on the limited set of 127 non-phishing sites seem to indicate that the concept does not produce a lot of false positives but this will definitely need a more detailed evaluation in the future. Especially comparing the search engine results with an own implementation of the Levensthein distance using a list of important domains might greatly improve the algorithm performance.

Besides this, the concept might even have the possibility to detect more websites than we were able to show with our evaluation. In the subdomain and pathdomain case, phishers had the freedom to place any fake URL they wanted – e.g. using "www.paypal.com.fake-domain.com". Our subdomain algorithm would then have used "www.paypal.com" as a query for the search engine, which does not trigger the spell checker as it is spelled correctly. Eventually, lots of other "correct" URLs would not even trigger the search engine's spell checker. Due to this fact we added the brand name checker to our algorithms. However this was just a very limited test with a small number of brand names. In summary the URL portions detected might be used for other means than spell checking only which again could greatly improve the performance of the concept.

Another issue of the presented concept could be the vast amount of web traffic to search engines that would be generated rolling out such a concept and eventual privacy issues caused by URL submission to a third party. For a production scale system, both problems could be solved. A special server architecture reduced to the components of word similarity and website importance would be enough to serve such requests. For better privacy, the detection algorithms could also run locally with a local reduced copy of the most important data from the online server.

7 Conclusions and Future Work

In this work we presented our concept of using URL similarity for the detection of fraudulent phishing websites. Using a search engine's spelling suggestions it is possible to validate various suspicious parts that can be extracted from possible phishing URLs. In a large quantitative evaluation with 8730 phishing URLs we were able to show that for the different extracted domain parts a noticeable number of phishing URLs triggered spelling suggestions. Cumulating all tests 54.3% of the websites would have triggered at least one of the tested features.

We additionally took screenshots of a subset of those attacks and measured their visual quality compared to their original. For those websites we were able to show that with rising perceived quality of the attacks the percentage of websites that trigger those cues rises too.

As future work, the testing methodology of our approach should be refined to be able to find properly spelled domains at other locations in the URL (e.g. the subdomain). Together with a better brand checker and a larger test set of original URLs the results should be confirmed. We recommend developing new tests for the URL parts that can be extracted. This could help validating correctly spelled domain terms at parts of the domain where they are unexpected. Also verifying the position and reasons for inclusion of certain brand names in the URL could help to detect fraudulent URLs. Besides this a reference implementation of the spell checking done by the search engine could help to tweak the algorithm performance for that specific domain of finding fraudulent URLs. In the end a field study with the concept rolled out to end users should be done.

8 Acknowledgments

We thank Alexander De Luca, Henri Palleis and Michael Rohs for their valuable input when writing this publication.

References

1. PhishTank: Statistics about phishing activity and PhishTank usage. http://www.phishtank.com/stats.php [last accessed 04/28/2012]
2. Goodin, D.: Google bots detect 9,500 new malicious websites every day. http://arstechnica.com/security/2012/06/google-detects-9500-new-malicious-websites-daily/ [last visited 07/12/2012]
3. Google Inc.: Safe browsing API — google developers. https://developers.google.com/safe-browsing/ [last accessed 04/28/2012]
4. Hong, J.: The state of phishing attacks. Communications of the ACM (2012)
5. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding phish: Evaluating anti-phishing tools. In: NDSS. (2007)
6. Moscaritolo, A.: Number of phishing URLs at alltime high. http://www.scmagazine.com/number-of-phishing-urls-at-all-time-high/article/150010/ [last visited 07/12/2012]
7. Riden, J.: How fast-flux server networks work. http://www.honeynet.org/node/132 [last visited 07/12/2012] (2008)
8. Whitten, A., Tygar, J.D.: Why johnny can't encrypt: A usability evaluation of PGP 5.0. In: 8th USENIX Security Symposium. (1999)
9. Dhamija, R., Tygar, J.D., Hearst, M.: Why phishing works. In: CHI. (2006)
10. Wu, M., Miller, R.C., Garfinkel, S.L.: Do security toolbars actually prevent phishing attacks? In: CHI. (2006)
11. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: Client-side defense against web-based identity theft. In: NDSS. (2004)
12. Zhang, Y., Hong, J.I., Cranor, L.F.: Cantina: a content-based approach to detecting phishing web sites. In: WWW. (2007)
13. Phelps, T.A., Wilensky, R.: Robust hyperlinks cost just five words each. Technical Report (2000)
14. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security (2011)
15. Krammer, V.: Phishing defense against IDN address spoofing attacks. In: PST. (2006)
16. Gabrilovich, E., Gontmakher, A.: The homograph attack. Communications of the ACM (2002)
17. Gusfield, D.: Algorithms on strings, trees, and sequences : computer science and computational biology. Cambridge University Press (1997)
18. Lin, E., Greenberg, S., Trotter, E., Ma, D., Aycock, J.: Does domain highlighting help people identify phishing sites? In: CHI. (2011)

19. Postel, J.: Domain Name System Structure and Delegation. RFC 1591 (Informational) (1994)
20. Mozilla Foundation: Public suffix list. http://publicsuffix.org/list/ [last accessed 04/29/2012]