# Rating Web Pages Using Page-Transition Evidence

Jian Mao[1], Xinshu Dong[2], Pei Li[1], Tao Wei[3], and Zhenkai Liang[2]

[1] School of Electronic and Information Engineering, BeiHang University
[2] School of Computing, National University of Singapore
[3] Institute of Computer Science and Technology, Peking University

**Abstract.** The rating of web pages is an important metric that has wide applications, such as web search and malicious page detection. Existing solutions for web page rating rely on either subjective opinions or overall page relationships. In this paper, we present a new solution, SnowEye, to decide the trust rating of web pages with evidence obtained from browsers. The intuition of our approach is that user-activated page transition behaviors provide dynamic evidence to evaluate the rating of web pages. We present an algorithm to rate web pages based on page transitions triggered by users. We prototyped our approach in the Google Chrome browser. Our evaluation through real-world websites and simulation supports our intuition and verifies the correctness of our approach.

## 1 Introduction

The rating of a web page is an important metric that has broad applications in the World Wide Web, ranging from web search to the detection of malicious web pages involving phishing or malware. A common way to rate web pages' trustworthiness is to rely on users to provide subjective opinions on web sites [1]. However, this is not a scalable approach. Moreover, the accuracy of such ratings depends on users' knowledge and experience, which might be biased.

Alternative solutions infer web page ratings using page relationships. For example, PageRank [2] used by Google leverages the link citation relationships among web pages to decide their ranking in search results. Such static link citations provide the necessary information to evaluate the overall popularity of web pages, yet they do not indicate the trustworthiness of these pages. Malicious pages could be cited frequently by another popular website, such as a web forum, which may give them relatively high rankings. PageRank also cannot access internal links, such as pages in a banking site that require customer login. Although such internal banking pages are highly trustworthy, they would usually have low ranking under algorithms such as PageRank.

**Our intuition:** In this paper, we leverage user-activated page transition behaviors to infer trust propagation among web pages. Our intuition is that page transitions resulting from user clicks play an important role in deciding the trustworthiness of the destination page given the trustworthiness of the source one. Such page transition behaviors provide dynamic evidence to derive more objective page ratings on their trustworthiness.

To understand the intuition, consider a new benign URL and a URL from a malicious site. The benign URL is usually visited by following links from other trustworthy pages. In contrast, malicious web sites are much less likely to be visited from links in

trustworthy web pages. Of course, page content from social networking sites and public web forums should not be considered as trustworthy, although such sites are not malicious. Therefore, the user-activated page transition to a web page is a form of evidence to indicate how trust propagates from the source site to the destination one. At the client side, web browsers can observe the dynamic behaviors of the non-public internal web pages and Ajax-based websites as well, long before they are available, if at all, in search engines.

Our intuition is based on an assumption that the vast majority of trustworthy web pages, such as banking pages, will not include links to arbitrary websites, and users are mindful in clicking links apparently leading to their intended destinations. It does not apply to cases where a large fraction of banking websites have already been compromised and thus containing links to malicious web pages. Addressing such devastating attack scenarios goes beyond the capability of rating-based security solutions.

**Our solution:** In this paper, we develop a novel approach, *SnowEye*, to evaluate the trust rating of web pages. The core component of our solution is a *dynamic-evidence-based algorithm* to quantify the rating of a web site. We use user-activated page transitions as the evidence to propagate the trust between original and destination web pages. Our solution propagates ratings from a (presumably small) pool of blacklisted and whitelisted web pages whose rating are pre-assigned by security experts. Note that malicious JavaScript can simulate user interactions with links in web pages; our approach is able to distinguish page transitions from such malicious scripts and those from genuine user behaviors, and only take page transitions by user clicks as the evidence for trust propagation.

We prototyped SnowEye in the Google Chrome browser, we evaluated it using a real-world dataset, and showed that our intuition is consistent with real-world scenarios.

In summary, we made the following contributions in this paper: *a)* we summarize the basic requirements for the page trust rating system based on page behaviors, *b)* we propose a novel algorithm to compute the trust of a web page based on dynamic evidences from browsers, and *c)* we prototype our solution in the Google Chrome browser and evaluated it with real-world data and simulation.

## 2 Problem & Approach

We define the problem we are targeting in this paper as follows.

**Problem 1** *Given a network of browsers, denoted as the set $\mathcal{N}$, using the set of past surfing behaviors $\mathcal{B} = \{b_i\}$ gathered from the browser set, how to calculate the trust rating of a target web page $Url_{Target}$, denoted as $R(Url_{Target})$?*

In this work, we focus on a basic scenario where we trust the browsers in reporting their behaviors. Note that a web page with high trust rating is conceptually different from a benign page. A benign page does not intentionally include malicious contents, or imitate other pages, but may include untrusted contents. Instead, a page with high trust rating indicates a benign page with probably only trusted content. Thus, web forums or social network sites that allow users to freely upload content may be benign sites, but their pages usually have low trust ratings.

### 2.1 Basic Requirements

In this paper, we treat browser page transition behaviors as the dynamic evidence to infer trust among web pages, where the source page transferring its trust to the target web page.

Given The *behavior* tuple $b = (Url_{original}, click, Url_{target}) \in B$, where the $Url_{original}$ means the original $Url$ before the event *click* and the $Url_{target}$ represents the $Url$ of the target page after *click*. the *rating* of the target page corresponding to this page transition triggered by the browser behavior $b$ should satisfy the following requirements:

*R1:* *Transition Property.* That means a web page transited from a high rating up-flow web page should be assigned a relatively high rating. At least, it should get the same rank as the original page.

*R2:* *Non-degradation Property*. This requirement claims that browser behaviors will not cause the decrease of the target page's rating.

*R3:* *Feedback Property*. If $Url_{Target} \in Blacklist$, then $R(Url_{Original})$ should be degraded sharply. The web page should be responsible for the trustworthiness of the links it cites. If a web page cites a fake page, it will be punished by our algorithm accordingly. To fulfill the property, the feedback function should be employed in the transition based rating algorithm.

The requirement *R1* is provided to match our algorithm's intuition that the page transition causes a rating transition consistent with its backward linked pages. Requirement *R2* is presented to prevent the *Malicious Citation Attack*, that is, malicious pages cite benign web pages to degrade the rating of the benign on. Even if there exist low rating websites citing the other pages in an attempt to degrade target page's rating, the benign page's rating will not be affected if it is pointed to by a relatively high rating web page (trustworthy page). In other words, the malicious citation is filtered out by the benign one. Requirement *R3* aims at the *False Citation Attack*, that is, attackers trick high-trust web sites to transit to malicious pages or arrange malicious websites to cite each other's pages to promote the rating of low-trust pages. It comes from the basic assumption that web page developers should audit the content inside their web sites carefully.

### 2.2 Rating Algorithm based on Dynamic Evidence

We use an iterative algorithm to compute the rating of the target URL based on the given behavior set $B$. The whole algorithm includes three parts: *Rating initialization*, *Rating computation*, and *Rating feedback & update*.

*Rating initialization* In our algorithm, we whitelist a set of pre-trusted web pages, blacklist a set of known malicious web pages, and we use them as a starting point to initialize the page rating. That is, if the target URL is a new page that has no rating before, it will get an initial rating $R^0$ according to the record of the whitelist and blacklist. If the target URL belongs to the whitelist, then it should be a good web page and gets the highest rating (e.g., 1). If the target URL belongs to the blacklist, then it should

be a malicious page and obtains the lowest rating (e.g.,0). If the page is published by a known developer who provided her/his technical confidence (or Reputation value, $t_{ic}$ for a node $N_i$) as the guarantee of the target page, then its rating value depends on the technical confidence of its publisher/developer and for some practical consideration, we use a relative technical confidence to initialize the rating of the new page. Otherwise, we assign a relatively small value $\delta$ (where $0 < \delta < \epsilon$, and $\epsilon$ is the threshold for the future decision making) to the new page as its initial rating.

$$R^0(url) = \begin{cases} 1, & \text{if } url \in whitelist; \\ 0, & \text{if } url \in blacklist; \\ \frac{t_{pc}}{max_{\{N_i\}} t_{ic}}, & \text{if } url \text{ is published by node } N_p; \\ 0 < \delta < \epsilon, & \text{Otherwise.} \end{cases}$$

*Rating computation* After the initialization part, The rating of the target $url$ confirmed by current page transition behavior $R^i(url|b_i)$ is quantified as follows

$R^i(url|b_i) = R^{i-1}(url_{original})$

Then we compute the final rating of the Target page $Url_{Target}$. The final rating of the $url$ should be the maximum value of behavior based ratings obtained so far corresponding to the target $url$.

$R^i(url) = max\{R^i(url|b_i), R^{i-1}(url)\}$

*Rating feedback and update* After getting the rating of the target page, if $R^i(url) < \epsilon$, where $\epsilon$ is a relatively low value depends on the effectiveness requirement, it means the target page might be a malicious page. It is necessary to review the target page and give a feedback to update (somehow, it means to degrade) the rating of its original link.

Then we set $R^i(url) = 0$ and $R^i(url_{original}) = R^{i-1}(url_{original}) \times e^{-k}$, where $k$ is the number of faulty citations found inside the web page $url_{original}$ by now.

From Algorithm 1 we can see that *Step 3* corresponds to the intuitive requirement $R1$ *Rating transition property*, in which web pages will get corresponding ratings according to page transition behaviors and at least obtain the same ratings as the original pages. *Step 4* corresponds to the intuitive requirement $R2$ *Non-degradation property* that new behavior will not cause the degrade of rating. Thus, if there exists a malicious citation attack in order to degrade the rating of the benign site, such a citation will be ignored. *Step 6* corresponds to the intuitive requirement $R3$ *Feedback property* that we will degrade the rating of a web page if it cites a malicious page.

## 3 Implementation and Evaluation

To evaluate the algorithms we propose for web page trustworthiness rating, we have prototyped SnowEye. In this section, we describe the implementation and evaluation results of our prototype.

### 3.1 Implementation

Our prototype of SnowEye uses a server to maintain a database of collaborative ratings to the hash values of target web site URLs, which are contributed from the browser

**Algorithm 1:** Basic Dynamic-Evidence-Based Rating Algorithm

**Input:** Page transition behavior $b = (Url_{original}, click, Url_{target})$, and Rating Database DB, which is initialized to the blacklist and whitelist.

**Output:** Trust rating of $R^i(Url_{target})$.

**Rating Initialization**

1: For $url$ = $b.d$=$Url_{Target} \notin DB_{url}$, computes

$$R^0(url) = \begin{cases} 1, & \text{if } url \in whitelist; \\ 0, & \text{if } url \in blacklist; \\ \frac{t_{pc}}{max_{\{N_i\}} t_{ic}}, & \text{if } url \text{ is published by } N_p; \\ 0 < \delta < \epsilon, & \text{Otherwise.} \end{cases}$$

**Rating compute**

2: Computes Target page rating confirmed by current behavior $b_i$
$R^i(url|b_i) = R^{i-1}(url_{original})$

3: Computes final rating of Target page $Url_{Target}$
$R^i(url) = max\{R^i(url|b_i), R^{i-1}(url)\}$

**Rating Feedback& Update**

4: Review the web page, if $Url_{Target} \in Blacklist$, then
$R^i(url) = 0$, and
$R^i(url_{org}) = R^{i-1}(url_{org}) \times e^{-k}$
Where, $k$ is the faulty citations found inside the $url_{original}$ by now.

5: Update the *Rating DataBase* $DB_{url}$.

6: Output $R^i(url)$

clients. The server records the ratings under each client's profile, and computes a client's technical confidence according to its rating history. The server also maintains a database of blacklists and whitelists based on existing solutions and manual reviews. The server is not necessarily a single node. It can be replaced by a set of distributed servers to enhance the responsiveness and reliability. In our current prototype, we implemented the server functionality in Perl.

We implemented the client side of SnowEye as an extension to Google Chrome. The extension monitors Chrome's internal behaviors about page transitions. The UI of the Chrome extension is implemented based on BlockUI [3]. In the extension, we inject content script to every page, which listens to page load and users' click behaviors and passes them to the extension core. When a page is loaded, the extension uses the corresponding page transition information to calculate the rating to the current page. If the rating is below the threshold, it will display a warning to the user. Optionally, such a warning can only be displayed when the extension detects that the current web page requests privacy information from users. In current implementation, the detection of web pages requesting privacy information is based on the heuristics that the web page contains `password` fields.

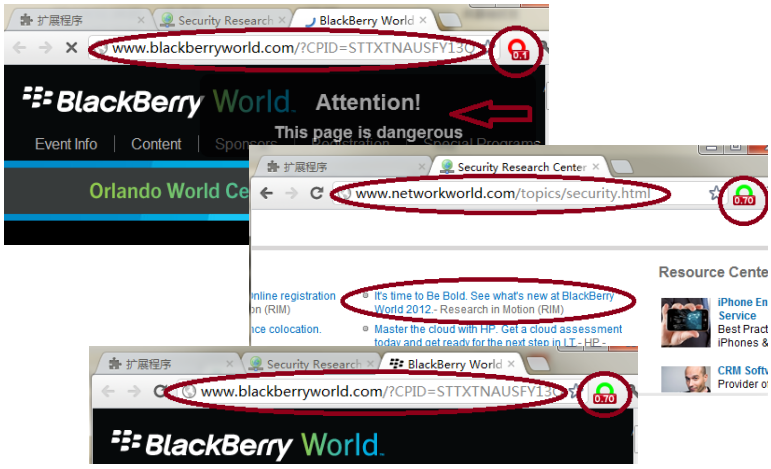


**Fig. 1.** A Sample Scenario

### 3.2 Experience with SnowEye

Now we present a sample working scenario of SnowEye, which demonstrates that web pages that were new to our system in the beginning would obtain higher trust values after being navigated to from a high trust pages. For example, in Figure 1, with a single client SnowEye browser, when the user first visits the page from *www.blackberryworld.com*, it only had the trust value 0.1, as SnowEye had no knowledge about this page, nor did it appear in the whitelist. But later when the user clicked on the link to it from the high-trust page *www.networkworld.com/topics/security.html*, according to Algorithm 1, such a click-triggered page transition propagated its trust value to the destination page, so *www.blackberryworld.com* also obtained the trust value 0.7.

### 3.3 Evaluation of Trust Propagation in URL Transitions

As our algorithm is based on one intuition that web pages with higher trust values are unlikely to link to malicious web pages, we performed the following experiment to verify this intuition with a phishing page as an example. Subsequently, we will also demonstrate how our algorithms automatically degrade the trust values of *"good"* web pages that link to malicious web pages.

We performed a crawling on 10 seed URLs shown in Table 1 by following the links on web pages we have crawled, and obtained 1388 unique URLs after 3-level crawling, i.e., web pages that were within 3 links away from the seed pages..

1. We further investigate the number of cross-site links from each of the seed site. Here cross-site links are loosely defined as links to a completely different domain other than the same domain or the subdomain of the original page. In another word, we use such cross-site links to investigate the links that cross the boundary of the seed web sites. Shown in Table 1, although all of the 10 seed web sites contain cross-site

| Seed Web Site | #Cross-site Links | #Malicious/Phishing |
|---|---|---|
| www.paypal.com | 253 | 0 |
| www.bankofamerica.com | 54 | 0 |
| www.chase.com | 55 | 0 |
| www.wellsfargo.com | 62 | 0 |
| www.americanexpress.com | 182 | 0 |
| www.hdfcbank.com | 18 | 0 |
| www.hsbc.com | 51 | 0 |
| www.citibank.com | 4 | 0 |
| www.capitalone.com | 7 | 0 |
| www.commbank.com.au | 32 | 0 |

**Table 1.** Seed Web Sites and the Number of Cross-site Links

links, none of the links point to a malicious or phishing web site. Therefore, as the seed sites serve as whitelist, according to our algorithm, all benign web sites linked from them will obtain good rating in our algorithm.

2. However, some of them point to benign web sites with untrusted contents, such as *www.twitter.com*, *www.google.com*, *www.facebook.com* and *myspace.com*, etc. In our crawling dataset, these sites will also obtain good ratings according to our algorithm. This is not expected as those sites contain untrusted content posted by users that may contain links to malicious or phishing web sites, and their high ratings may be transferred to those malicious or phishing web sites. To solve this issue, we simulate the rating feedback mechanism in our algorithm by introducing another dataset, which is obtained by performing backward searching for web pages citing phishing pages. This verified that bringing the rating feedback is critical in our algorithms. As these benign sites also have links to phishing sites [4], our algorithm degraded their trust values, although they may obtain high values in the beginning.

### 3.4 Trust Value Degrading

Now we show a concrete example of how SnowEye degrades web page originally with high trust values after their links to phishing web pages are detected. As illustrated in Figure 2, in the beginning, the page had a high trust value 0.80. However, it had links to three phishing pages. When later the user clicked on the first link, such a page transition was captured by SnowEye. As the destination page was in the phishing page blacklist, the destination page was given the trust value 0.00. Besides, according to the feedback step in Algorithm 1, this page transition also degraded the trust value of the page *blog.sina.com.cn/s/blog_75b798e501012vb7.html* according to number of faulty citations detected in that page. So in this round, its trust value was degraded to $0.80 * e^{-1} = 0.29$. Similarly, when the user clicked on the second link, as it also redirected to a phishing page, the trust value of *blog.sina.com.cn/s/blog_75b798e501012vb7.html* was further degraded to $0.29 * e^{-2} = 0.04$. In the figure, we omitted the similar step after the user clicked on the third link, which degraded the trust value of *blog.sina.com.cn/s/blog_75b798e501012vb7.html* to


**Fig. 2.** Degrading Trust Values of Pages Linking to Phising Pages

0.00. This example demonstrated that the basic algorithm of SnowEye automatically degrades the trust values of pages that made faulty citations to phishing pages.

## 4 Related Work

### 4.1 Page Relationship Based Solutions

Google uses PageRank [2], a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them. It produces a global "*importance*" ranking of every web page based on the link structure. Actually, the simplicity of creating and publishing web pages results a large fraction of low quality web pages that user are unlikely to read. Attackers can make use of this to promote the rating of their malicious pages. This kind of approaches based on static linkage are not suitable to evaluate the trust rating of the web pages.

### 4.2 Subjective Feedback Based Solutions

WOT [1] and iTrustPage [5,6], provide the solutions to rate the phishing possibility of a given page by using reputation scores either reported from the anti-phishing community or computed from the given web page. However, these two approaches are user-assisted. WOT's rating algorithm is based on the comments subjectively submitted by the users.

Nettrust [7] tries to connect people with each other by social network. In their framework, people make some interaction with the server to get some sharing experience (some kind of subjective comments and ranks) released by other nodes/members. The approach tries to defend the attack based on some kinds of common sense among the social peers. Nettrust proposes an e-mail based model to establish a security-related community, and mentions some privacy problems briefly. But there is no practical model or approach was presented to defend a specific attack and does concern about the malicious information posted by tricky members. The final score is the average of negative ranks and positive ranks separately.

### 4.3 Webpage Feature Based Solutions

SpoofGuard [8] is a browser plug-in that users domain name, URL, link and image check to determine if a given page is a part of a spoof attack. It applies three methods and combines the result using a scoring mechanism: a stateless method to determine whether a downloaded page is suspicious; a stateful method to evaluate downloaded page in the light of user's history and a method to evaluate outgoing HTML post data.

CANTINA [9] uses a content-based approach to detect phishing web sites. It combines a term frequency-inverse document frequency (TF-IDF) algorithm with other heuristics to determine whether a given web site is phishing one. The method uses five words with the highest TF-IDF weight on a given web sites as a signature and then submits those five words to the Google search engine. If the URL of the site is found within top results, then that URL is classified as legitimate, otherwise phishing. An attacker could bypass CANTINA using several approaches. Such as, use image instead of words in a given page, add invisible text that is tiny or matches background color of the page, or change a lot of words in order to confuse TF-IDF.

### 4.4 Black/White-list solutions

EBay Toolbar [10] The eBay toolbar is an extension to Microsoft Internet Explorer and combines a tool named $AccountGuard$ to protect against spoofed eBay or PayPal web sites. The tool can identify if any particular URL which is trying to phish ebay.com, but it cannot handle the phishing URL targeting some other web sites. Bayesian Anti-phishing toolbar [11] is a whitelist based approach using DOM analyzer to check if the given URL is a legitimate web-site listed in the whitelist. If the URL is not in the pre-set whitelist, DOM analyzer labels the given web site and sends it to a scoring module. If the score exceeds a selected threshold, the URL is classified as malicious. Cloudmark [12] rates the web sites based on the system maintained blacklist. When the user surfs to the malicious web-site in blacklist, Cloudmark will direct the user to a specific page that illustrates the security risk. The blacklist database is partly maintained by the users in the way that user can report the malicious web-site. The system also contains a user rating scheme based on users' behavior to prevent users submitting fake report. the blacklist and the rating database are managed and audited manually. The effectiveness totally depends on the users and the system operators' experience and honesty.

Unlike the web page rating solutions discussed above, SnowEye monitors the dynamic features of web page transitions in the browser. The corresponding information is treated as critical metrics in web page trust rating evaluation. In addition, this objective information is verifiable and cannot be forged. The trust rating generated by SnowEye is more reliable. Attackers may mimic the static features of a web page somehow (e.g., URL, web page layout, content, etc.), but they cannot forge users' surfing behavior successfully without being detected.

## 5 Conclusion

In this paper, we present a novel trust rating approach for web pages, which is based on dynamic evidence captured by web browsers. Our approach, SnowEye, treats user-activated page transitions as an objective and dynamic evidence for the trust rating of web pages. Based on this intuition, we developed an algorithm to compute the suspicious ratings of the target web-pages. We prototyped our approach in the Google Chrome browser and evaluated it using real-world examples and simulation. Our evaluation verified our intuition and showed the effectiveness of SnowEye.

**Acknowledgment.** The authors thank anonymous reviewers for their insightful comments. This work was supported in part by the Beijing Natural Science Foundation (No. 4132056), the National Key Basic Research Program (NKBRP) (973 Program) (No. 2012CB315905), the Beijing Natural Science Foundation (No.4122024), and the National Natural Science Foundation of China (No. 61272501, 61173154, 61003214), and the Ministry of Education of Singapore via Tier-1 grant R-252-000-460-112.

## References

1. WOT, http://www.mywot.com.
2. L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999.
3. BlockUI, "jquery blockui plugin," http://jquery.malsup.com/block/.
4. "Report: Bank of melbournes twitter feed used for phishing," http://www.thetechherald.com/article.php/201138/7633/Report-Bank-of-Melbourne-s-Twitter-feed-used-for-Phishing.
5. T. Ronda, S. Saroiu, and A. Wolman, "itrustpage: A user-assisted anti-phishing tool," in *Proceedings of Eurosys'08*. ACM, April 2008.
6. iTrustPage, http://www.cs.toronto.edu/~ronda/itrustpage/.
7. L. J. Camp, "Net trust: Signaling malicious web sites," 2007.
8. D. Boneh, "Spoofguard (2011)," http://crypto.stanford.edu/SpoofGuard.
9. Y. Zhang, J. Hong, and L. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in *Proceedings of the International World Wide Web Conference (WWW)*, May 2007.
10. eBay Inc., "ebay toolar (2011)," http://www.pages.ebay.com/ebay_toolbar/.
11. L. P., E. Jung, D. D., H. T.E., and H. J.P., "B-apt: Bayesian anti-phishing toolbar," in *Proceedings of IEEE International Conference on Communications, ICC'08*. IEEE Press, May 2008.
12. C.Inc., "Couldmark toolbar," http://www.cloudmark.com/desktop/ie-toolbar.