SURF: Detecting and Measuring Search Poisoning

Long Lu
College of Computing
Georgia Inst. of Technology
long@cc.gatech.edu

Roberto Perdisci
Dept. of Computer Science
University of Georgia
perdisci@cs.uga.edu

Wenke Lee
College of Computing
Georgia Inst. of Technology
wenke@cc.gatech.edu

ABSTRACT

Search engine optimization (SEO) techniques are often abused to promote websites among search results. This is a practice known as *blackhat SEO*. In this paper we tackle a newly emerging and especially aggressive class of blackhat SEO, namely *search poisoning*. Unlike other blackhat SEO techniques, which typically attempt to promote a website's ranking only under a limited set of search keywords relevant to the website's content, search poisoning techniques disregard any term relevance constraint and are employed to poison popular search keywords with the sole purpose of diverting large numbers of users to short-lived traffic-hungry websites for malicious purposes.

To accurately detect search poisoning cases, we designed a novel detection system called SURF. SURF runs as a browser component to extract a number of robust (i.e., difficult to evade) detection features from *search-then-visit* browsing sessions, and is able to accurately classify malicious search user redirections resulted from user clicking on poisoned search results. Our evaluation on real-world search poisoning instances shows that SURF can achieve a detection rate of 99.1% at a false positive rate of 0.9%. Furthermore, we applied SURF to analyze a large dataset of search-related browsing sessions collected over a period of seven months starting in September 2010. Through this long-term measurement study we were able to reveal new trends and interesting patterns related to a great variety of poisoning cases, thus contributing to a better understanding of the prevalence and gravity of the search poisoning problem.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval—*Relevance feedback*

General Terms

Security

Keywords

Search engine poisoning, Malicious search engine redirection, Detection, Measurement

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CCS'11, October 17–21, 2011, Chicago, Illinois, USA.
Copyright 2011 ACM 978-1-4503-0948-6/11/10 ...$10.00.

1. INTRODUCTION

Search engines, capable of digging out the most relevant from oceans of information, have become web surfers' first choice when seeking information on the web. In fact, for most websites more than 70% of their visitors reach their pages through search engines [6]. Therefore, website owners always strive to attract more visits by optimizing their exposure in relevant search results. To fulfill this need, web developers use a number of search engine optimization (SEO) techniques, which can improve the visibility of a website to the search crawlers, highlight its relevance under certain search terms, and promote its raking in the search results.

Legitimate uses of SEO techniques are accepted and even encouraged by search engines [1]. However, dishonest web developers may choose to abuse these techniques in various ways to gain (or cheat) a favorable ranking in the search results, a practice known as *blackhat SEO*. In this case, search crawlers are presented with deceptive views of a website, which consist of specially crafted webpages with inflated relevance to a set of target search terms. Attempts to counter blackhat SEO have been proposed mainly in the information retrieval community [18, 24], but with very limited success against the recent surge of blackhat SEO adopters [11]. In the meantime, blackhat SEO has not captured sufficient attention from the security community, perhaps because such techniques have been historically employed by non-harmful websites, including some high profile ones [10], that execute overly aggressive marketing strategies to win search users from their competitors.

This paper tackles a newly emerging class of blackhat SEO techniques developed by Internet miscreants to lure search users into visiting malicious websites [7]. We refer to this new class of blackhat SEO as *search poisoning*. Unlike other blackhat SEO techniques, which typically attempt to promote a website's ranking only under a limited set of search keywords relevant to the website's content, search poisoning techniques disregard any term relevance constraint. In practice, search poisoning techniques target any search term that can maximize the number of incoming search users (e.g., popular keywords). This is in contrast with SEO or other blackhat SEO techniques adopted by regular websites, because if search poisoning were to be used to promote a regular website, users landing on the website via completely unrelated search terms may get annoyed and the website's reputation may be irreparably damaged. Therefore, we posit that search poisoning cannot be used for legitimate purposes and is only useful to short-lived traffic-hungry websites that aim to attract search users for malicious purposes.

We approach the search poisoning problem from a new angle, compared to previous work on blackhat SEO. We focus on detecting *malicious search user redirections*, an essential component of

search poisoning that we discovered during our study (Section 2.2). To detect malicious search user redirections, we designed a novel detection system named **SURF** (Search User Redirection Finder), which runs as a browser component and is able to accurately detect poisoned search results. In this paper, we restrict our definition of "malicious search user redirections" to be any redirection that starts from a search landing page (i.e., the immediate page a search result points to) promoted by search poisoning and ends at an unwanted or malicious terminal page (i.e., the final destination of the redirection). We show that these malicious redirections can be characterized by a number of distinctive features that are collectively difficult to circumvent. SURF is designed to capture this very type of malicious redirections, and in turn to detect search poisoning instances. Other malicious redirections on landing pages not involving search poisoning are out of the scope of this work.

To detect search poisoning instances, SURF analyzes data collected during a *search-then-visit* browsing session and outputs a classification result, namely whether or not the monitored browsing session includes malicious search user redirections caused by a poisoned search result. As shown in Figure 1, the monitored browsing session starts from the search result page, passes by the landing page and possibly some intermediate pages, and stops at the terminal page to which the search user is eventually redirected. During this course, SURF collects the following information: (i) browser events to track page (and frame) loads and redirections; (ii) network information to model the redirection chain; and (iii) search result information to measure the poisoning likelihood of the landing page. From this information, SURF extracts a number of statistical features, which are then fed to a classifier trained to identify instances of search poisoning. Given the adversarial nature of classifying search poisoning instances, we selected our feature set with particular emphasis on individual feature *robustness* while retaining collective feature *generality*. To identify a suitable set of features, we performed a manual analysis of a dataset $\mathcal{S}_{study}$ containing 1,084 real-world search poisoning cases (see Section 2.2). We then experimented with 15 candidate features on a separate dataset $\mathcal{S}_{eval}$ containing 2,344 independently labeled samples of browsing sessions, 1,160 of which where related to search poisoning and there remaining 1,184 were not. Through feature selection, we refined the initial feature set down to 9 features based on which we evaluated SURF's classifier, which achieved an average true positive rate of 99.1% at a false positive rate of 0.9%.

To further demonstrate the effectiveness of SURF, we developed a prototype system and deployed it to classify daily popular search results obtained from both Google and Bing over a period of seven months (from September 2010 to April 2011). This large scale evaluation allowed us to conduct extensive measurements of real-world search poisoning cases, through which we observed that, on average, about 50% of popular searches contain poisoned results. During this measurement study we also observed a surge in both the volume of search poisoning instances and the variety of malicious content reached through poisoned search results.

In summary, this paper makes the following contributions:

- We clearly define search poisoning and distinguish it from other abuses of SEO techniques. We report an in-depth study to motivate and inspire countermeasures against this increasing threat.

- We design and evaluate SURF, a system that is able to detect search poisoning with a 99.1% true positive rate at a 0.9% false positive rate.

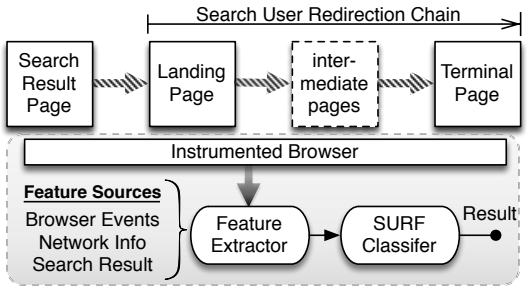

Figure 1: SURF Architecture

- Using a SURF prototype, we conducted a seven-month measurement study on poisoned popular search results, which highlights the severity of search poisoning and provides insight into its fast growing trends.

2. BACKGROUND AND PROBLEM STUDY

In this Section, we provide a brief overview of the fundamentals of search engines and the reason why search results are subject to manipulation. We then present a study of search poisoning based on real world data, and discuss the observations that inspired our detection approach.

2.1 Search Engine and Blackhat SEO

Search engines typically employ crawlers to discover newly created or updated webpages. Each crawled page is then indexed based on keywords retrieved from its content. Upon a search query, webpages are ranked based on their relevance to the search terms and presented to the user.

This gives the abusers the following advantages. First, search engines implicitly trust the authenticity of the content on the indexed webpages, even though the content is under complete control of the website owners. Second, a web server can easily distinguish between search crawlers and human visitors. It is this *implicit trust* and *distinguishability* that give rise to blackhat SEO, whereby a web server fulfills webpage requests from crawlers with specially crafted content having inflated relevance to a selected set of keywords, referred to as the *target keywords*. In addition, these crafted pages often contain large numbers of cross-reference hyperlinks to webpages that belong to the same blackhat SEO campaign. These hyperlinks have the effect of increasing the incoming link count for the promoted webpages, thus boosting the ranking in the search results.

Despite the fact that they involve some level of dishonesty, blackhat SEO techniques are sometimes used by legitimate businesses. In this paper, we distinguish between two types of blackhat SEO, namely *search inflating* and *search poisoning*. Search inflating aims to boost a website ranking through search keywords closely related to the promoted website, therefore only attracting users who search for topics related to the promoted website. On the other hand, search poisoning aims to boost a website's ranking though popular search keywords, regardless of whether these keywords are actually related to the promoted website's content or not. Unlike search inflating, which may be adopted by some legitimate websites, search poisoning only fits the need of visitor-hungry websites that simply want to increase the number of visitors for malicious purposes (e.g., for malware propagation purposes).

Most previous works on blackhat SEO detection apply lexical and structural analysis on page content [18, 22], or graph-based analysis on hyper-links [24]. However, very limited success has been achieved in practice [9], as the battle soon turned into an arms

race. Having full control over webpage visibility and the freedom of adopting new evasion techniques gives adversaries significant upper hands, and makes it fundamentally difficult to design robust detection schemes based on lexical and structural analysis. To mitigate these problems, our approach aims to detect search poisoning instances using a set of features that are collectively difficult to evade because they are intrinsic to how search poisoning works, as we discuss in Section 3.2.1.

2.2 Search Poisoning Study

Search poisoning instances luring visitors to malware websites were first reported in 2007 [7]. However, until recently search poisoning has not been sufficiently studied, and has been only sporadically mentioned within the anti-malware community (mostly due to *fake AV* websites [20]).

To gain a more in-depth understanding of the search poisoning problem, we manually analyzed a dataset $\mathcal{S}_{study}$ containing 1,084 real-world search poisoning cases collected in September 2010. This preliminary study aimed to discover a set of robust features that can be leveraged for detection purposes, and to inspire our overall detection approach.

To collect the dataset $\mathcal{S}_{study}$ we proceeded as follows. We deployed an army of instrumented browsers, which on a daily basis automatically query Google and Bing with keywords that have been popular for the past 7 days. For each query, the browsers visited the top 100 URLs in the search results[1]. All network data and browsing events occurred during each browsing session were recorded as a browsing trace. This data collection process resulted in a very large dataset $\mathcal{D}$ containing over half a million browsing traces. $\mathcal{S}_{study}$ was derived from $\mathcal{D}$ using a simple heuristic to select traces that lead to malicious [3] or non-reputable webpages [4] with content irrelevant to the search keywords. This coarse-grained filtering yielded 1,084 highly likely search poisoning cases consisting of 596 unique landing URLs. It is worth noting that this dataset is not meant to be inclusive of all poisoning traces in $\mathcal{D}$, which is impossible to achieve without first developing a reliable detection system. However, our filtering heuristic produced a $\mathcal{S}_{study}$ dataset that exhibits satisfactory accuracy and sufficient diversity, as confirmed by our manual analysis. Therefore, $\mathcal{S}_{study}$ represents a reasonable base for our preliminary study of search poisoning.

Below we itemize our observations and lessons learned from our manual analysis, which inspired the choice of the statistical features used by SURF.

O1: Ubiquitous use of cross-site redirections

We found that over 98% traces in $\mathcal{S}_{study}$ contain one or more redirections that cross website boundaries. The remaining 2% of browsing traces that do not contain such redirections are mostly due to incompletely rendered webpages, or modal dialogs that require non-trivial user interactions to proceed. On the other hand, less than 6% of the entire traces in $(\mathcal{D} - \mathcal{S}_{study})$ involve cross-site redirections. This ubiquitous use of cross-site redirections can be intuitively explained by the high risk and low effectiveness of exposing the malicious terminal website directly to search engines for rank promotion (thus a separated landing website is needed). For example, search engines have various security detectors in place to filter known malicious webpages and downgrade ranks of suspicious ones. Therefore, directly promoting malicious webpages can be a vain attempt and risks to jeopardize the entire search poisoning campaign. In fact, as our study went deeper, more evidence emerged supporting the need for malicious search user redirection in search poisoning.

[1] excluding URLs already flagged as malicious by search engines.

O2: Search poisoning as a service

From all traces in S_{study} we extracted their chain of redirections, which are then used to compose a redirection graph, in which the nodes represent encountered domains and the directed edges represent redirections from one domain to another. Large numbers of inter-connected chains form subgraphs that represent different search poisoning campaigns. Two representative subgraphs are shown in Figure 2, as examples to illustrate our findings. Successful campaigns are able to employ many landing domains and target different search keywords to maximize the incoming search users. Figure 2 (top) shows a campaign that successfully poisoned over 28 "trendy" search keywords and injected at least 46 URLs into top search results. Furthermore, the variety of terminal domains supported by a single campaign suggests that specialized search poisoning services are available to all kinds of malicious websites for purchase. The graph also indicates a two-tier affiliate marketing model followed by this campaign. Some landing pages redirected search users to centralized "super affiliate" domains (circled in the graph), which then dynamically dispatch the lured users to different terminal domains. As a result, more intermediate webpages appeared in the redirection chains.

O3: Sophisticated poisoning and evasion tricks

Cloaking techniques [12] are commonly used in search poisoning (by 97% landing pages in S_{study}). Search crawlers are presented with specially crafted content with fake relevance. The malicious redirection process only starts when visited by search users that queried the target (poisoned) keywords, while blocking other visitors as an attempt to prevent security detectors reaching the malicious content or domains. By forging the browser's `User-Agent` strings, we managed to obtain the search crawler views of 26 landing pages in our S_{study} dataset that did not verify the crawler's source IP address. These views were carefully composed to mimic normal webpages (e.g., blogs or news sites), with highly relevant content (possibly scraped from elsewhere) organized in a smooth way. We noticed that this well-crafted content may easily fool human readers, and is therefore very likely to evade content-based blackhat SEO detectors. In addition, we discovered a handful of image-rich landing pages that likely targeted at multimedia searches.

Another way in which search poisoning try to evade detection is by hosting the landing pages on compromised websites. These websites typically have been indexed by search engines for quite some time and accumulated a non-trivial domain history or reputation. This can help search poisoning to bypass some security checks performed by search engines and to facilitate rank promotion. In our study dataset S_{study} we found that about 70% of the redirections start from domains with a fair reputation score in [4] and only 2% originate from blacklisted domains in [3].

O4: Persistence under transient appearances

To achieve a persistent poisoning effect, search poisoners have to accommodate for the volatility of popular search keywords. The bottom graph in Figure 2 shows a campaign that made multiple appearances on different popular search results across the entire study period: old landing domains had been active only for a limited time (a few days) before a new batch came in with a new set of poisoned search keywords. Rapidly rotating landing domains not only enable a wide coverage on trendy search topics, but also hinder detection efforts due to their transient appearance. Terminal domains behave in a similar way. An important difference is the fact that terminal domains tend to be disposable and have short registration periods (likely using domain tasting services), which further impedes blacklist-based detection.

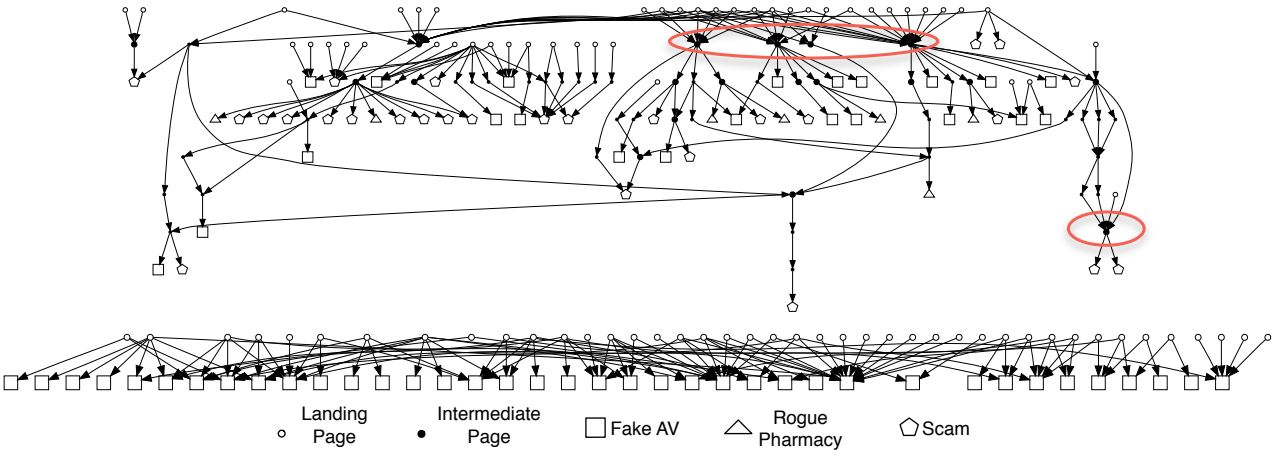

Figure 2: Redirection graphs of two search poisoning campaigns

O5: Various malicious applications

While previous reports always associate search poisoning with malware distribution websites [8, 14], search poisoning is used in a variety of other malicious applications. Figure 2 shows at least three types of malicious websites that use search poisoning to promote different types of cyber crimes, such as distributing fake AV software, hosting of rogue pharmacy sites, and other types of scams. Other types of uses (not reported in Figure 2) were also observed in S_{study}, such as sites that host drive-by download exploits, are part of click fraud schemes, or host phishing pages. Therefore, we argue that security solutions specific to individual types of malicious websites fail to thoroughly address the general search poisoning problem.

Lessons learned: Detecting search poisoning is a daunting task. Solely relying on identifying suspicious features associated with the landing pages (e.g., deceptive relevance, suspicious linkage structures, etc.) is immediately subject to evasion, given the attackers' freedom to craft the page content. At the same time, detecting malicious terminal pages is hindered by their diversity and conditional accessibility of the actual malicious content (in particular, malicious sites can detect crawlers and security scanners). However, our observations convey a positive message to the defenders: the malicious search user redirections are intrinsic to search poisoning cases and exhibit distinguishable behaviors that are difficult to avoid completely for search poisoning to be successful, as we further discuss in Section 3.2.1. SURF's design was inspired by these findings.

3. SURF DESIGN AND EVALUATION

Based on the lessons learned from our search poisoning study (Section 2.2), we set three primary design goals for our detection system:

- *Generality*: Search poisoning techniques are employed by attackers to promote a variety of malicious contents, and are not limited to luring users to visiting malware distribution pages. Therefore, SURF aims to detect generic search poisoning instances, regardless of the malicious content the attackers intend to promote.
- *Robustness*: While it is arguably impossible to completely prevent an arms race between defenders and attackers, we aim to identify features typical of search poisoning cases that are difficult to evade. In practice, we restrict SURF to using a set of *robust* features, which cannot be evaded by adversaries without incurring a significant cost (e.g., because of the need to completely change their attack strategy, or move to a different attack infrastructure).
- *Wide deployability*: Unlike most previous work on blackhat SEO detection, which is constrained by the dependency on search engine private data and only deployable at the search engine side, our approach aims to provide a solution that can be deployed at end-user's browsers (as a plugin), at automated security crawlers, and inside search engines. End-users can be protected from malicious terminal webpages hidden behind poisoned search results. At the same time, search engines or security vendors can deploy SURF in a "browsers farm" to accurately detect whether a search keyword is poisoned and harvest the malicious terminal page.

3.1 SURF System Overview

To meet the goals listed above, we designed SURF as a browser component. An overview of SURF is shown in Figure 1. In practice, SURF sits in a browser and observes search-related browsing sessions. Whenever a user submits a query to a search engine and receives the result page, SURF starts its monitoring on page loads and redirections, from the search result page, to the landing page (on user's click), and to the terminal page the user is eventually brought to (after going through several intermediate pages in some cases). During this course, SURF extracts a number of statistical features from a range of sources, such as browser events, network information regarding the domain names and IP addresses involved in the redirection chain, and the search results themselves (see Section 3.2 for details). The resulting feature vector is then sent to the *SURF Classifier*, which is trained to distinguish between normal redirections and *malicious search user redirections*. In practice, the *SURF Classifier* is a supervised statistical classifier trained using a labeled dataset containing examples of redirection chains resulted from clicking on either legitimate or a variety of poisoned search results. It is worth noting that our definition of "malicious search user redirections" in this paper is restricted to redirections following poisoned search results. Detecting other types of malicious redirection is out of the scope of this work. While the vast majority of redirections used by the search poisoning cases we encountered during our study (Section 2.2) only change the URLs of webpages' top frames, SURF also covers the malicious redirections that occur within dynamic subframes (e.g., an `iframe`).

3.2 Detection Features

Inspired by the study presented in Section 2.2, and keeping in mind the design goals outlined at the beginning of Section 3, we

Aspects	Features	Source*	Evasion Possibility
Redirection Composition	Total redirection hops	B	Low
	Cross-site redirection hops	B,N	Low
	Redirection consistency	B	Low
Chained Webpages	Landing to terminal distance	B,N	Low
	Page load/render errors	B,N	Low
	IP-to-name ratio	B	Medium
Poisoning Resistance	Keyword poison resistance	S	Low
	Search rank	S	Low
	Good rank confidence	S	Low

* B=browser events/data; N=network info; S=search result

Table 1: Feature selection

identified a set of nine statistical features that capture the characteristics of generic search poisoning instances, and that are difficult for the attacker to evade without incurring a significant cost (e.g., the attacker would need to move to a new search poisoning strategy and infrastructure). The features extracted by SURF are divided in three groups as summarized in Table 1 and detailed below.

Redirection composition: This group of three features aims at capturing discrepancies between the legitimate and malicious search user redirections. The `total redirection hops` records the number of redirections that transport the visitor from the landing page to a terminal page, whereas the `cross-site redirection hops` counts how many of these redirections cross website boundaries. In SURF, a cross-site redirection is a redirection that brings from a domain d_1 to a domain d_2, where the *second-level* domains[2] of d_1 and d_2 differ (e.g., `www.cnn.com` and `blogs.cnn.com` share the same second-level domain, while `www.cnn.com` and `www.bbc.com` do not). As noted in Section 2.2, the vast majority of poisoned search results rely on cross-site redirections to transport search users to the malicious terminal pages hosted on covert domains. On the other hand, legitimate search user redirections rarely send incoming visitors away to another websites, simply because of the common incentive of keeping as many visitors as possible within their own domain. The `redirection consistency` feature captures whether a redirection is only visible to targeted search users. In legitimate search user redirections, users who arrive to the landing page through a search engine or though a direct link (e.g., by typing the same URL on the browser's address bar, or clicking the hyperlink on a "non-search" website) will be redirected to the same terminal page displaying relevant content. This is in contrast with search poisoning cases, in which typically only users that reach the landing page through a search will be redirected to the malicious content, while other visitors will be presented with non-malicious content in an attempt to evade some detection systems or manual analysis (see Section 2.2). To measure redirection consistency, SURF can first command the browser to directly visit the landing page (thus effectively stripping the `Referrer` field in the HTTP request, for example), and then allows the user's click on the search result link to go through so that the two obtained sets of redirection events can be compared for consistency. When used at the search engine-level (e.g., in a "browsers farm"), SURF could perform this comparison on redirections obtained by visiting the landing page from different IP addresses to bypass some of the cloaking techniques discussed in Section 2.2 whereby the malicious content is not provided to IP addresses visiting the same landing page twice in a row. It is worth noting that such cloaking technique would not affect SURF's protection when deployed at end user's browsers. If the attacker refuses to offer the malicious content at the second visit (i.e., when SURF allows the user's click on a search result to go through), the user will not be exposed to the malicious content in the first place.

Chained webpages: This group of features measures three properties of the webpages involved in search user redirections. The `landing to terminal distance` feature measures the (approximate) geographical and topological distances between the landing page and the terminal page. In practice, to measure this distance we leverage information about the geo-location of the IPs where the two pages reside, the autonomous systems (AS) the IPs belong to, and the websites' domain names. The intuition is that malicious search user redirections always "travel" a long distance. The reason is that in search poisoning cases the landing page is usually hosted on a (likely compromised) website that belongs to a separate (usually legitimate) organization, while the terminal page is often hosted on a "bullet-proof" server provided by a different (usually not legitimate or boarder-line) organization often located in a different country.

The `page load/render errors` flags pages in the redirection chain that failed to load or render properly, due to exceptions or network errors. The intuition is that compromised pages are sometimes blacklisted or remediated, and the redirection chain to the malicious terminal page may end prematurely. The `IP-to-name ratio` feature represents the number of the redirection URLs that use an IP address (e.g., `http://192.168.0.1/index.php`) divided by the number of redirection URLs that instead use a domain name (e.g., `http://example.com/index.php`). This is motivated by the fact that a large number of search poisoning cases encountered in our study involve URLs that use IPs that are dynamically assigned to unnamed hosts, in an effort to bypass URL blacklists commonly available in major browsers.

Poisoning resistance: This group of features measure properties of the search keywords and their corresponding search results. The `keyword poison resistance` quantifies the difficulty of poisoning search results under a given keyword. We measure this feature using publicly available information. The basic idea is straightforward: given a certain search keyword, the competitiveness of promoting a link higher in the result rankings is reflected by how prominent the top ranked webpages are under that keyword. We use Google's PageRank [19] to measure the prominence of a website. The `keyword poison resistance` of a keyword k is defined as the average PageRank value of the top 10 ranked websites obtained from the search results under k. In practice, the higher the value of this feature, the more prominent websites competing for the top rank positions, and thus the more difficult for an attacker to poison the keyword and force a link to a rogue landing page to appear higher in the ranking.

From our study dataset $\mathcal{S}_{study}$ (see Section 2.2), we noticed that the distribution of poisoned keywords across different search poisoning cases is skewed towards keywords with low `keyword poison resistance`. This result was somewhat expected, because keywords that are popular and yet easier to poison than others tend to attract the attackers' attention. Another feature we consider is the `search rank` of a landing page. The higher the rank of a landing page, the lower the probability that the result has been poisoned. This follows directly from our previous argument that top ranked pages are often prominent websites, making it difficult for search poisoners to promote their sites ahead of these prominent sites. The `rank confidence` feature is computed by dividing the `keyword poison resistance` by the rank of a partic-

[2]The second-level domain of a domain name `d.c.b.a` is typically defined as `b.a`, where `a` is called the *top-level* domain. We leverage Mozilla's public domain suffix list to take effective top-level domains such as `co.uk` into account.

ular search result. The higher the `rank confidence`, the less likely that the result is poisoned.

3.2.1 Qualitative Robustness Analysis

Here we present a qualitative analysis of the robustness of SURF's statistical features against evasion attempts. A quantitative robustness analysis is discussed in Section 3.4.

Redirection composition: This group of features tries to capture the "search poisoning as a service" phenomenon discussed in Section 2.2. In practice, attackers often compromise several legitimate websites to host rogue landing pages, which use deceptive content to promote their ranks and at the same time redirect lured search users to the malicious terminal webpages. While in principle an attacker can attempt to evade this group of features, this would force the attacker to move to a completely different malicious network infrastructure in which all malicious redirections and terminal malicious pages are hosted on the same second-level domain, for example. In addition to incurring the significant cost of changing the malicious network infrastructure, this could also increase the risk of being detected by the compromised website's administrators, thus exposing search poisoning instances to a more prompt remediation, and end up sacrificing the established rogue landing pages which typically take considerable amount of time and efforts to mature.

Chained webpages: This group of features measure properties of the webpages involved in search user redirections. The `IP-to-name ratio`, while useful for classification in most practical cases, is not very difficult to evade because it only requires the attacker to register more domain names. However, we would like to emphasize that carrying an evasion attack on this feature will not significantly impact SURF's accuracy, as shown in the quantitative analysis in Section 3.4. The remaining two features are harder to evade. The `landing to terminal distance` feature depends on the attacker's network infrastructure, and therefore the same arguments we made for the *redirection composition* features hold in this case. The `page load/render errors` may depend, for example, on pages (and domains) along the redirection chain that have been blacklisted or remediated. Therefore, this feature is not under the attacker's direct control and is difficult to evade.

Poisoning resistance: Since these three features are derived from the search result pages themselves with the help of public PageRank data, which is determined by the search engine algorithms and out of attackers' control. Therefore these features are difficult to evade.

3.3 Prototype Implementation

SURF only requires a limited amount of data to be collected during *search-then-visit* browsing sessions, and can be easily incorporated into a browser as a plugin. To demonstrate the effectiveness of our detection approach, we implemented SURF on top of an instrumented version of Internet Explorer 8. This instrumented browser leverages `mshtml.dll` for HTML parsing and rendering, and is able to listen for event notifications (e.g., used to identify subframe redirections) and peek into browsing data (e.g., HTML code and user visible content) at different rendering stages. In addition, SURF is capable of emulating simple user interactions during visiting sessions that require certain user input to proceed (e.g., clicking on a message dialogue box, activating a mouse-over action, etc.).

To perform our evaluation of SURF, we used several instrumented browser instances and instructed them to retrieve lists of popular (or "trendy") search keywords, query each keyword on both Google and Bing, and visit the top 100 search results for each query. Our evaluation data collection started in September 2010. We deployed our browser on 20 virtual machines which would run

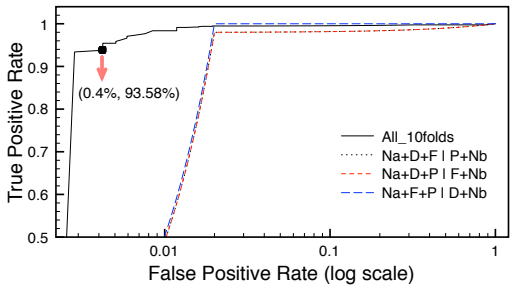

Figure 3: Threshold Curves (ROC)

daily to query search keywords that had reached popularity within the past seven days. In addition, our instrumented browser was enhanced with BLADE [16], which we used to protect the VMs from *drive-by download* malware infections and to log attempted exploits. The obtained browsing information for each browsing session was then saved for offline analysis. While we performed our evaluation offline for convenience reasons, mainly to be able to run cross-validation experiments on large datasets, once the statistical classifier has been trained SURF can detect search poisoning instance *online* (see Figure 1).

3.4 Evaluation Results

Evaluation Dataset: To the best of our knowledge, there exist no public labeled dataset of search results and their related user redirection events related to search poisoning cases. Therefore, we chose to semi-manually label part of our dataset. In particular, we labeled browsing sessions collected during October 2010. In the following, we refer to obtain dataset as $\mathcal{S}_{eval}$. It is worth noting that $\mathcal{S}_{eval}$ differs from the $\mathcal{S}_{study}$ dataset that we used for the search poisoning study in Section 2.2 ($\mathcal{S}_{study}$ was collected one month earlier). This is to make sure that SURF does not "overfit" $\mathcal{S}_{study}$ because $\mathcal{S}_{study}$ inspired the choice of SURF's statistical features, and to avoid over-estimating SURF's accuracy. For the same reasons, none of the statistical features measured by SURF were used to guide our labeling. We simply semi-manually labeled the data using a separate set of heuristics based on the relation between the search terms and a (visual) analysis of the terminal page content rendered by the browser.

In practice, we labeled search poisoning cases related to three types of malicious activities: search result for popular search keywords that lead to irrelevant terminal pages serving (1) *drive-by download* malware, (2) *fake AV* software, or (3) hosting *rogue pharmacy* sites. We labeled all these cases as *poisoned* (or positive). At the same time, we labeled a search result as *non-poisoned* (or negative) only when all URLs appeared in redirection chain have a fair reputations (e.g., according to [4]) and none of them are flagged as malicious by website scanning or blacklisting services (e.g, using [3]). Overall, $\mathcal{S}_{eval}$ consists of 1,184 negative samples and 1,160 positive ones, with 585 fake AV, 414 drive-by download, and 161 rogue pharmacy cases.

To evaluate SURF and confirm it follows the design goals outlined at the beginning of Section 3, we conducted three different experiments. The first experiment aims to estimate SURF's accuracy, while the second attempts to show that SURF is able to detect generic search poisoning cases, and is not limited to one specific type of malicious content. The third experiment aims to show what features are the most important for classification, and how SURF may respond to evasion attempts on these features. Throughout all our experiments, we used Weka's `J48` decision tree classifier [13], which is an implementation of the well known C4.5 algorithm [5]. This choice is motivated by the fact that decision trees are efficient

(both during the training and testing phases) compared to other statistical classifiers, and the resulting trained classifier can be easily interpreted.

Overall accuracy: To estimate SURF's detection rate and false positives, we performed a 10-fold cross validated of SURF's classifier on the $\mathcal{S}_{eval}$ dataset, achieving an average true positive rate of 99.1% at a 0.9% false positive rate. The ROC curve in Figure 3 shows a very slow decrease of the detection rate as the false positive rate is pushed down from 0.9% to 0.28%. Therefore SURF can satisfy a relatively wide range of usage scenarios with different levels of tolerance to false positives, while still maintaining a reasonably high true positive rate. In our SURF prototype we set the detection threshold to limit the false positive rate to 0.4% (marked point in Figure 3). We also analyzed the decision tree produced by the trained `J48` classifier and found that misclassifications were mainly caused by those rare cases in which poisoned search results achieved a top rank under a very competitive keyword, or cases in which a legitimate landing page redirects visitors to a very "distant" terminal page (see Section 3.2) and detours sampled visitors through third-party traffic analysis services.

Generality test: To confirm that SURF is able to detect generic search poisoning cases, regardless of the specific malicious content that they promote, we performed the following experiment. We prepared three datasets, $\mathcal{D}$, $\mathcal{F}$, and $\mathcal{P}$, containing labeled search poisoning example from the drive-by malware downloads, fake AV, and rogue pharmacy cases from $\mathcal{S}_{eval}$, respectively. In addition, we prepared two separate datasets, $\mathcal{N}_a$ and $\mathcal{N}_b$, containing randomly selected negative examples (i.e., legitimate search redirection cases). We then performed a 3-fold cross validation by using $\mathcal{D} \cup \mathcal{F} \cup \mathcal{N}_a$ for training, and testing on $\mathcal{P} \cup \mathcal{N}_b$ for the first fold, training on $\mathcal{D} \cup \mathcal{P} \cup \mathcal{N}_a$ and testing on $\mathcal{F} \cup \mathcal{N}_b$ for the second fold, and training on $\mathcal{F} \cup \mathcal{P} \cup \mathcal{N}_a$ and testing on $\mathcal{D} \cup \mathcal{N}_b$ for the third one. Figure 3 shows the three separate ROC curves (one per fold). Averaging the results of the 3-fold validation, we obtained a detection rate of 98% at a false positive rate of 1.8%. It is worth noting that while this result does not appear to be as good as the result obtained for the overall 10-fold cross validation experiment, our 3-fold evaluation presents SURF with much harder cases in which no examples of search poisoning instances of the same category used for testing are present in the training set. However, the high detection rate and relatively limited false positives demonstrate that the features used by SURF can indeed capture generic search poisoning properties, independent of the specific type of malicious content delivered by the terminal page.

Feature Robustness: This experiment attempts to quantify SURF's resistance to evasion effects on the statistical features. To perform the experiments, we ran SURF's classifier on 100 randomly chosen positive samples. We artificially modified the values of the features describing these 100 samples to simulate different evasion scenarios, and evaluated how the classifier's accuracy changed as a consequence. We first artificially set the `IP-to-name ratio` to zero, which is the most common value of the feature in negative samples. The `IP-to-name ratio` feature is the only one among SURF's features that is not difficult to evade. After altering this feature, only 1 out of 100 samples was misclassified, which suggests that the attacker cannot gain much by attempting to evade it. Despite the fact that other features are hard to evade (see Section 3.2.1), we nonetheless wanted to investigate the effect of altering the most discriminant features, i.e., the features that appeared close to the root of the decision tree. The `redirection consistency` and `landing to terminal distance` turned out to be the top two most discriminant features. Replacing their values with values drawn from negative samples caused 80 out of 100 samples to be misclassified. However, it is worth noting that evading these two features would require the attacker to change her malicious network infrastructure, thus incurring a significant cost, as discussed in Section 3.2.1.

4. DISCUSSION

At a first glance, our evaluation may seem unconvincing because we used a dataset labeled by ourselves. However, to the best of our knowledge, no public labeled dataset exists that contains browsing session data related to a variety of search poisoning instances. As discussed in Section 3.4, our semi-manual labeling process is based on a set of heuristics that do not overlap with any of the detection features measured by SURF. This allowed us to perform an unbiased evaluation. While we acknowledge that our semi-manual labeling can only help us collect partial "ground truth", it is extremely difficult, if not impossible, to obtain complete ground truth without a deterministic search poisoning detection system. In absence of such perfect search poisoning detector, our labeling method represents our best effort to produce a representative (even though not complete) ground truth that includes a variety of search poisoning instances leading to different types of malicious content.

During our feature selection process, we discarded a few candidate features that may help the classification accuracy but are not robust. For example, we chose not to include features based on measuring the relevance of the content of terminal pages to the search keywords because the content of the terminal pages is under complete control of the attacker, making these types of features easy to evade. Also, we did not include features related to the structure of the URLs involved in malicious search user redirections. These features usually require historical knowledge of the "normal" structure of the URLs for each particular website. While these type of features may be included in a search engine-side deployment of SURF, client-side deployments would not be able to collect and leverage this kind of information, and therefore we decided not to add them to our prototype implementation. Furthermore, an attacker has a non-negligible flexibility when choosing the structure of the redirection URLs, and therefore it is not clear how robust these features would be. We also considered some features based on domain or IP reputation scores. Though capable of reducing the false positive rate, these features were excluded from our selection because of their heavy dependence on external security services, and because we wanted to evaluate the detection accuracy of SURF based solely on the strengths of our own features. In practice, SURF implementations may opt to incorporate reputation-based features to improve classification accuracy.

When deployed at the search engine side (e.g., using a "browser farm"), SURF can be used to analyze suspicious search results and accurately detect poisoning cases. This can provide search engines with valuable information that goes beyond specific poisoning instances. For instance, the landing pages involved in search poisoning are often organized in a "botnet mode", so that the keywords to be poisoned can be periodically fetched from a command-and-control server. Therefore, detecting search poisoning cases can reveal information about compromised websites and botnet organizations. In addition, newly detected malicious terminal webpages may serve as labeled samples for malicious webpage detectors that require periodic re-training.

At the same time, SURF can be deployed at each single client to detect (and block) poisoned search results. A possible deployment scenario could include large numbers of client-side SURF installations that collaboratively detect search poisoning case and share information about the underlying malicious network infrastructures

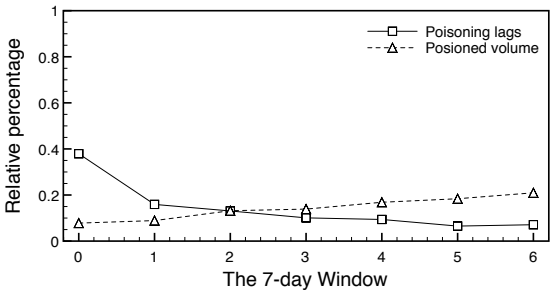

Figure 4: Micro measurement statistics

(e.g., domain names, IP address, etc.) through a cloud service, thus potentially improving SURF's detection accuracy.

5. EMPIRICAL MEASUREMENTS

To gain a deeper insight into the search poisoning problem, we performed a long-term (7 months, 212 days) measurement study. As discussed in Section 2, we manually analyzed data in the first month. We used SURF to analyze browsing data we collected in the next 6 months. During this period we instructed SURF to analyze over 12 millions search results from both Google and Bing collected by querying the top 40 "trendy" keywords [2]. In practice, once a search keyword appeared in the top 40 list on a given day, we used SURF to query the search engines for this keyword for the following 7 days. Overall, during our measurement study SURF automatically queried the search engines with 8,480 keywords. This long-term data collection process enabled us to study in-the-wild search poisoning instances from two different angles, a "micro measurement" study based on a 7-day window that focuses on how search poisoning evolves with respect to frequently changing keyword popularity, and a "macro measurement" study that looks at poisoning trends over the entire 7-month period.

5.1 Micro Measurements

Due to frequent changes in the trendy search keywords [2], we expected that some time (e.g., a few days) would be required for the attackers to poison the search terms and make their landing pages of choice appear in the related top search results. Surprisingly, our measurements on the micro developments of detected search poisoning cases suggest otherwise: adversaries are extremely responsive and have built effective approaches to promote rogue landing pages under the targeted trendy search keywords in a short time.

Among the 3,869 keywords for which we detected related poisoned search results, 38% of them had *poisoning lag* (i.e., the time it takes for the first poisoned result to be detected) of one day or less. This percentage decreases as the lag increases, and only 7% of the keywords had a poisoning lag of 7 days, as shown in Figure 4. This results suggest that the adversaries are capable of keeping up with search users' interests, and that the majority of their poisoning attempts succeed within the first 3 days.

We also found that the average life time of a rogue landing page involved in search poisoning is only 1.7 days. This indicates that adversaries favor a fast-switching strategy to reduce the exposure window, thus reducing the possibility of the rogue pages being detected and conserving the compromised landing sites for reuse in future poisoning attacks. However, the appearance of these rogue landing pages in the search results lasts for more than 3 days on average, until the page ranking is demoted due to the new information retrieved by the search crawlers. At the same time, the relative volume of detected rogue landing pages for a given poisoned keyword keeps increasing throughout the 7-day observation window, as shown by the *poisoned volume* increase in Figure 4 (the poisoned volume is relative to the total number of detected poisoned results during the 7-day period). We believe this fast paced operation proposes significant challenges for blacklisting and other traditional security solutions, making them inadequate to solve the search poisoning problem. In fact, well-known malicious webpage scanners (e.g., [3]) have failed to detect 78.9% of malicious terminal pages involved in the search poisoning cases detected by SURF (we scanned all the terminal pages using [3] on the same day when SURF detected the related search poisoning instance).

Our measurements show that the visiting traffic reaching a particular malicious terminal page is always contributed by multiple landing pages that appear in poisoned search results related to different search keywords. In particular, we found that during a 7-day period, for each given malicious terminal page, their visitors were supplied in average by 2.9 poisoned search keywords and 2.2 different rogue landing sites per keyword. We speculate this is a reflection of a growth in the "search poisoning as a service" phenomenon discussed in Section 2.2.

5.2 Macro Measurements

After examining the development of poisoned search results in the first 7 days period, we now zoom out our measurement window to consider the entire 7-month data collection period. Through these long-term measurements we aim to discover search poisoning's evolving trends and characteristics that are observable only throughout a long period of time. To highlight specific patterns and long-term trends, we divide the 7-month observation period in 31 epochs, where each epoch is equal to one week. Then, for each week we compute a number of statistics (discussed below), and plot how these statistics vary with respect to time.

During most epochs, we found that more than 50% of the search keywords that became popular on that epoch got poisoned. For this particular measurement, a search keyword is considered to be poisoned if at least one out of the top 100 search results for that keyword is related to a search poisoning instance. Figure 5 (left) plots the percentage of poisoned keywords for each week, broken down into different degrees of success in terms of search ranking. We can see that during some of the epochs, almost 60% of the trendy search keywords resulted in rogue landing pages that ranked within the top 50 search results. Furthermore, in some cases adversaries managed to promote their rogue landing pages up to the top 10 search results. These particularly successful attacks were related to about 15% of all poisoned keywords on average. These findings are alarming because they suggest that a large number of search users can easily be affected by search poisoning.

Figure 5 (right) shows two curves. One represents the number of rogue landing sites (counted as the number of distinct related domain names) that were involved in search poisoning cases, and the other the number of distinct (*landing domain*, *keyword*) pairs. The two peeks (marked by A and B in the figure) that appeared around Christmas time and the Super Bowl are not a coincidence. Our analysis shows that important predictable events can help adversaries to further increase their poisoning success rate, given the sufficiently large preparing time (enabled by the predictability of the events) and the interest of large number of search users in the events being exploited. At the same time, less predictable breaking news that receive broad attention for a not too short amount of time (e.g., a few days) are also an easy target for search poisoning. An example of this is the earthquake and tsunami that recently hit Japan (marked by C in Figure 5). Furthermore, as the targeted keywords (upper curve) fluctuated between attempts to leverage different events, the number of detected landing domains (lower curve) remained somewhat more stable, suggesting that search poisoning

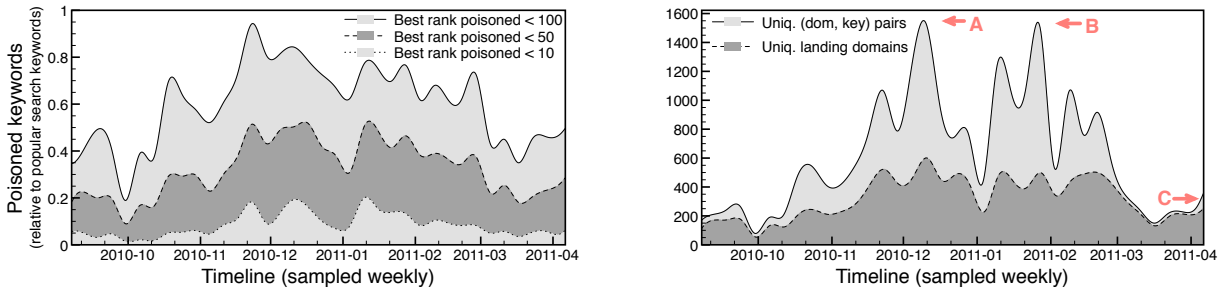

Figure 5: Poisoned keywords percentage (left) and landing domains engaging search poisoning (right)

operators have a solid footing in the search engines' indexes, and are ready to launch new attacks whenever the opportunity comes.

On average, users who fall victim of search poisoning are redirected at least twice, including one cross-domain redirection, before reaching the malicious terminal page. About 29% of these redirections were due to `HTTP` 30x responses. Not surprisingly, the majority of the remaining 70% were mostly due to client-side scripts (likely an attempt to evade security crawlers that do not support script execution). About 78% of the landing page URLs explicitly contained the targeted search keywords to boost the page relevance perceived by search crawlers. About 98% of the intermediate URLs include ID-like parameters to track unique visitors, identify search poisoning affiliates, or prevent repeated visits. In 94% of the cases, the terminal page URLs used domain names registered for less than a year, with many of these domain chosen to purposely deceive victim users and promote specific scams. Among the detected search poisoning cases, the most frequently used top-level domains (TLDs) for landing pages are `.com`, `.org`, `.net`, and `.info`, with a TLD distribution similar to regular websites. On the other hand, domains related to terminal pages were mostly registered under TLDs such as `.cc`, `.com`, `.in`, and `.net`, some of which are known to be malware friendly.

To have a sense of the variety of malicious content promoted by search poisoning, we surveyed 350 randomly chosen terminal pages. These 350 terminal pages were evenly distributed across our 7-months measurement period. The results are shown in Figure 6. We manually categorized the terminal pages based on data saved at the time when the page was visited by SURF, including screen shots of each rendered page. As expected, fake AVs are the most prevalent adopters of search poisoning during the entire 7-month period. However, we noticed that their pervasiveness started fading as other types of malicious terminal pages increased. Drive-by malware downloads and other browser exploitation techniques did not appear to be commonly leveraging search poisoning. On the other hand, various types of social engineering-based malicious pages are dominant players. The figure shows a clear surge of rogue search engine pages, which present the users with links seemingly relevant to the search keyword but aim to profit from user clicks.

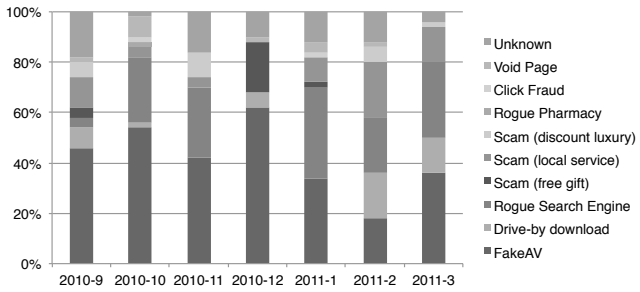

Figure 6: Terminal page variety survey

Scam pages (e.g., watch replicas, etc.) represent another significant fraction of the surveyed terminal pages. Regardless of their individual tactics, scam pages in general bait traps with free or unrealistically cheap goods to attract users and steal private information (e.g., credit card numbers, passwords, etc.). We also encountered a number of malicious terminal pages related to click fraud and rogue pharmacies. The terminal pages categorized as "void" typically contain clues of of certain types of maliciousness (e.g., based on their domain name patterns) but were inaccessible when visited due to unsuccessful DNS resolution, or webpage errors. SURF's ability to detect even these "void" malicious terminal pages supports our initial goal of building a detection system that is agnostic to the specific content of malicious pages promoted through search poisoning.

6. RELATED WORK

In this section, we identify and discuss two lines of work related to our detection system.

Blackhat SEO countermeasures: Blackhat SEO, which involves abusing search engine optimization techniques to achieve undeserved rankings, is not a new problem and has been studied for year, especially in the information retrieval community. Most proposed detection methods work at the search engine level and attempt to identify deceptive information introduced by the adversaries into the search index to influence the rankings of their websites. Various detection features explored by these methods mainly focus on two aspects of indexed webpages: intra-page characteristics [18,22,25] and inter-page linkages [24]. However, for adversaries with full control over their injected search landing pages, such features are not difficult to evade, sometimes even without requiring changes to their operation routines. In fact, this traditional way of countering blackhat SEO has failed to stop its rising trend [11]. SURF addresses search poisoning, a new class of blackhat SEO, building on the lessons learned from previous work and approaching the detection from a new angle using a set of feature that is more robust to evasion (see Section 3.2.1).

deSEO [15] is a very recent work done in parallel with SURF. It detects URLs from the search index that contain signatures derived from known search poisoning landing pages and exhibit patterns not previously seen by the search engine on the same domain. Since there is no need to crawl each URL, this approach scales much better than SURF when facing a huge volume of search results. However, deSEO is limited by the coverage of the URL signatures, and may only find a subset of what SURF detects. For example, about 12% landing page URLs detected by SURF in Section 5.2 do not contain trendy search keywords, and thus may be missed by deSEO. Moreover, SURF does not rely on any information internal to search engines and can be deployed at the client side, enabling single browsers to detect poisoned search results as well as malicious webpages behind them before the content is presented to the user.

Malicious webpage detection: SURF, when implemented as an automated detection agent, can be viewed as a dynamic crawler used to scan search results looking for poisoned ones. From this perspective, SURF is similar to many proposed systems that crawl the Internet for various kinds of malicious webpages [17,23]. Such systems always employ an army of browsing agents running in a controlled environment to visit suspicious URLs in batch and detect signs of specific types of malicious activities. SURF can be easily integrated into these systems and can enable the detection of search poisoning cases along with the related compromised landing page and malicious terminal pages. On the other hand, solely relying on malicious page detectors for finding poisoned search results may achieve limited success, because of the variety of terminal pages, many of which use social engineering attacks that are difficult to detect.

Applying machine learning techniques to data collected during a crawling session is also a common approach to detecting malicious webpages. A recent work [21] is able to detect URLs that lead to spam pages. Our work is different because SURF is not limited to detecting spam pages, and can instead detect generic search poisoning cases.

7. CONCLUSION

Search poisoning is an abuse of SEO techniques by which miscreants target any search term that can maximize the number of incoming search users to their malicious websites. We observed through an empirical study that a key characteristic of search poisoning is the ubiquitous use of cross-site redirections. We designed and implemented SURF, a novel detection system that runs as a browser component and is able to detect malicious search user redirections resulted from user clicking on poisoned search results. SURF extracts a number of detection features from *search-then-visit* browsing sessions. These features are robust and the resulting classifier is hard to evade because they capture the key properties of search poisoning (derived from our empirical study and analysis). Our evaluation showed that SURF can achieve a detection rate of 99.1% at a false positive rate of 0.9% on a dataset that contains real-world search poisoning instances. Using SURF, we also performed a long-term measurement study on search poisoning on the Internet over a period of seven months. Our study revealed new trends and interesting patterns related to a great variety of poisoning cases, and underscored the prevalence and gravity of the search poisoning problem.

8. ACKNOLEDGEMENT

The authors would like to thank the anonymous reviewers for helpful comments on earlier versions of the paper. This material is based upon work supported in part by the National Science Foundation under grant no. 0831300, the Department of Homeland Security under contract no. FA8750-08-2-0141, the Office of Naval Research under grants no. N000140710907 and no. N000140911042. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Department of Homeland Security, or the Office of Naval Research.

9. REFERENCES

[1] Google search engine optimization. http://www.google.com/webmasters/.

[2] Google trends. http://www.google.com/trends.

[3] URLVoid: Scan a website with multiple scanning engines. http://www.urlvoid.com/.

[4] WOT: Web of trust. http://www.mywot.com/wiki/API.

[5] *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[6] Google drives 70 percent of traffic to most web sites. http://searchengineoptimism.com/Google_refers_70_percent.html, July 2006.

[7] Malware poisoning results for innocent searches. http://www.eweek.com/c/a/Security/Malware-Poisoning-Results-for-Innocent-Searches, November 2007.

[8] Barracuda labs 2010 mid-year security report. Technical report, Barracuda Networks Inc., 2010.

[9] Search engine optimization 'poisoning' way up this year. http://www.networkworld.com/news/2010/110910-seo-poisoning-increases.html, November 2010.

[10] The dirty little secrets of search. http://www.nytimes.com/2011/02/13/business/13search.html, February 2011.

[11] Google: Search engine spam on the rise. http://www.pcworld.com/article/217370/google_search, January 2011.

[12] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. Technical report, Stanford University, 2005.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[14] F. Howard and O. Komili. Poisoned search results: How hackers have automated search engine poisoning attacks to distribute malware. Technical report, SophosLab, 2010.

[15] J. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy. deSEO: Combating search-result poisoning. In *Proceedings of the 20th USENIX Security*, 2011.

[16] L. Lu, V. Yegneswaran, P. Porras, and W. Lee. Blade: an attack-agnostic approach for preventing drive-by malware infections. In *Proceedings of the 17th ACM CCS*, 2010.

[17] E. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy. A crawler-based study of spyware on the web. In *Proceedings of the NDSS*, 2006.

[18] A. Ntoulas and M. Manasse. Detecting spam web pages through content analysis. In *In Proceedings of the 15th WWW*, 2006.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[20] M. A. Rajab, L. Ballard, P. Mavrommatis, N. Provos, and X. Zhao. The nocebo effect on the web: an analysis of fake anti-virus distribution. In *Proceedings of the 3rd USENIX LEET*, 2010.

[21] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *In Proceedings of the IEEE S&P*, 2011.

[22] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking web spam with html style similarities. *ACM Trans. Web*, 2:3:1–3:28, March 2008.

[23] Y.-M. Wang, D. Beck, X. Jiang, and R. Roussev. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *Proceedings of the NDSS*, 2006.

[24] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th WWW*, 2005.

[25] B. Wu and B. D. Davison. Detecting semantic cloaking on the web. In *Proceedings of the 15th WWW*, 2006.