PoisonAmplifier: A Guided Approach of Discovering Compromised Websites through Reversing Search Poisoning Attacks

Jialong Zhang, Chao Yang, Zhaoyan Xu, Guofei Gu

SUCCESS Lab, Texas A&M University
{jialong, yangchao, z0x0427, guofei}@cse.tamu.edu

Abstract. Through injecting dynamic script codes into compromised websites, attackers have widely launched search poisoning attacks to achieve their malicious goals, such as spreading spam or scams, distributing malware and launching drive-by download attacks. While most current related work focuses on measuring or detecting specific search poisoning attacks in the crawled dataset, it is also meaningful to design an effective approach to find more compromised websites on the Internet that have been utilized by attackers to launch search poisoning attacks, because those compromised websites essentially become an important component in the search poisoning attack chain.

In this paper, we present an active and efficient approach, named PoisonAmplifier, to find compromised websites through tracking down search poisoning attacks. Particularly, starting from a small seed set of known compromised websites that are utilized to launch search poisoning attacks, PoisonAmplifier can recursively find more compromised websites by analyzing poisoned webpages' special terms and links, and exploring compromised web sites' vulnerabilities. Through our 1 month evaluation, PoisonAmplifier can quickly collect around 75K unique compromised websites by starting from 252 verified compromised websites within first 7 days and continue to find 827 new compromised websites on a daily basis thereafter.

1 Introduction

Search Engine Optimization (SEO) manipulation, also known as "black hat" SEO, has been widely used by spammers, scammers and other types of attackers to make their spam/malicious websites come up in top search results of popular search engines. Search poisoning attacks, as one particular type of "black hat" SEO, inject malicious scripts into compromised web sites and mislead victims to malicious websites by taking advantages of users' trust on search results from popular search engines. By launching search poisoning attacks, attackers can achieve their malicious goals such as spreading spam, distributing malware (e.g., fake AntiVirus tools), and selling illicit pharmacy [14]. For example, in April 2011, many search terms (e.g., those related to the royal wedding between Britain Prince William and Catherine Middleton) are poisoned with Fake AntiVirus links [15]. These links mislead victims to install fake Security Shield AntiVirus software. In 2011, one research group from Carnegie Mellon University also reported substantial manipulation of search results to promote unauthorized pharmacies by attackers through launching search poisoning attacks [3].

Essentially, search poisoning attacks compromise benign websites by injecting malicious scripts either into existing benign webpages or into newly created malicious pages. Then, these scripts usually make the compromised websites respond with different web content to

users that visit via or not via particular search engines. Specifically, once the compromised websites recognize that the requests are referred from specific search engines, the compromised websites may lead the users to malicious websites through multiple additional redirection hops. However, if the compromised websites recognize that the requests are directly from users, the compromised websites will return normal content rather than malicious content. Thus, this kind of cloaking makes the attack very stealthy and difficult to be noticed. In addition, the good reputation of these (compromised) websites (e.g., many are reputable .edu domains) essentially help boost the search engine ranks and access opportunities of malicious webpages. In this case, it is meaningful to discover those compromised websites as many as possible to stop such search poisoning attacks.

Most current state-of-the-art approaches to find such compromised websites merely utilize pre-selected key terms such as "Google Trends [6]", "Twitter Trending Topics [16]" or specific "spam words" to search on popular search engines. However, the number of newly discovered compromised websites by using this kind of approaches is highly restricted to those pre-selected key terms. First, the limited number of the pre-selected terms will restrict the number of compromised websites that could be found. Second, since these terms usually belong to some specific semantic topics, it will be hard to find more compromised websites in different categories. In addition, since many pre-selected key terms (e.g., Google Trends) are also widely used in benign websites, such approaches will also search out many benign websites leading to low efficiency.

In this paper, we propose a novel and efficient approach, PoisonAmplifier, to find compromised websites on the Internet that are utilized by attackers to launch search poisoning attacks. Specifically, PoisonAmplifier consists of five major components: Seed Collector, Promote Content Extractor, Term Amplifier, Link Amplifier, and Vulnerability Amplifier. Seed Collector initially collects a small seed set of compromised websites by searching a small number of terms on popular search engines. Then, for each known compromised website, Promote Content Extractor will extract "promoted web content", which is promoted by compromised website *exclusively* to search engine bots, but not seen by normal users. This web content is essentially promoted by attackers and usually has close semantic meaning with final malicious website (e.g, illicit pharmacy content). Through extracting specific query terms from "promoted web content", Term Amplifier will find more compromised websites by searching those query terms instead of simply using pre-selected key terms. The intuition behind designing this component is that attackers tend to provide similar key terms for search engine bots to index the webpages. For each compromised website, Link Amplifier first extracts two types of links: inner-links and outer-links. Inner-links refer to those links/URLs in the promoted web content of the compromised website. Outer-links refer to those links/URLs in the web content of other websites, which also have link of known compromised website. Then, Link Amplifier finds more compromised web sites by searching those inner-links and outer-links. The intuition is that the links in the promoted content tend to link to other compromised websites. Also, the websites linking to known compromised websites may also link to other (unknown) compromised websites. Vulnerability Amplifier will find more compromised websites, which have similar system or software vulnerabilities to existing known compromised websites. The intuition is that attackers tend to exploit similar vulnerabilities to compromise websites to launch search poisoning attacks. Through implementing a prototype system, PoisonAmplifier, our approach can find around 75,000 compromised web sites by starting from 252 known comprised websites within first 7 days and continue to find 827 new compromised websites

everyday on average thereafter. In addition, our approach can achieve a high Amplifying Rate[1], much higher than existing work [22, 23].

The major contributions of this paper can be summarized as follows.

- We propose PoisonAmplifier, a new, active, and efficient approach to find more compromised websites by analyzing attackers' promoted content and exploiting common vulnerabilities utilized by the attackers. Rather than simply using pre-selected (static) keywords to search on popular search engines, PoisonAmplifier is more effective and efficient to discover more compromised websites.
- We implement a prototype system and evaluate it on real-world data. Through our evaluation, PoisonAmplifier can find around 75,000 compromised websites by starting from only 252 verified compromised websites within first 7 days.[2] As a comparison with two recent studies using pre-selected terms, it takes 9 months to collect 63K compromised websites in [22] and 1 month to collect 1K compromised websites in [23]. Furthermore, PoisonAmplifier can discover around 4 times and 34 times compromised websites by analyzing the same number of websites, compared with [22] and [23].

The rest of the paper is structured as follows. We describe the background and our targeted search poisoning attacks in Section 2. We present the whole system design of PoisonAmplifier in Section 3 and the evaluation in Section 4. We discuss our limitations in Section 5 and current related work in Section 6. Finally, we conclude our work in Section 7.

2 Background

In this Section, we first provide a brief overview on how our targeted search poisoning attacks work. Then, we present two typical methods utilized by attackers to promote malicious content in search engines to launch such search poisoning attacks.

2.1 Threat Model

Search Engine Optimization (SEO) is the process of optimizing websites to have higher search engine ranks. It includs white hat SEO and black hat SEO. Unlike white hat SEO, which increases search ranks by constructing websites to be more easily crawled by search engines, black hat SEO techniques attempt to obtain high rankings by violating search engines' policies such as keyword stuffing [12], hiding texts [9], and cloaking [2].

In our work, we focus on one specific type of black hat SEO techniques, named Search Poisoning Attack, which usually responds with malicious content to the users referred via search engines, while responds with non-malicious content to the direct visiting users. Next, we will describe how this search poisoning attack works.

As illustrated in Figure 1, to launch such a search poisoning attack, an attacker typically needs to first compromise a website by exploiting the website's system/software vulnerabilities, and then injects malicious scripts (PHP or Javascript) into the compromised website (labeled as ① in Figure 1). The core of such search poisoning attack is the ability for the compromised website to utilize injected malicious scripts to recognize different origins of the requests. Specifically, once the compromised website finds that the requests originate from crawler bots such as Google Bots, the website responds with web content containing special

[1] It is the ratio of the number of newly discovered compromised websites to the number of seed compromised websites.

[2] This speed is limited by the search rate constraint imposed by the Google search engine.

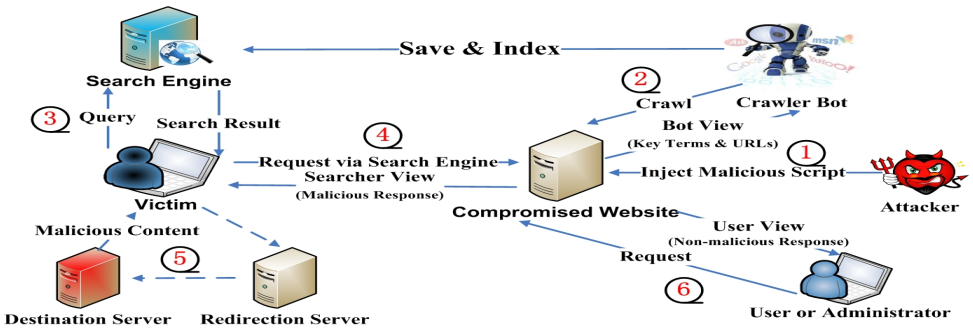


Fig. 1. The work flow of search poisoning attacks.

keywords and URLs injected by attackers. These special keywords and URLs are essentially what attackers desire to promote exclusively to the search engine crawler bots and hope to be indexed by the search engines(②). Then, if a user queries those keywords on search engines (③) and sends requests to the compromised website by clicking on the search results, the user will be a desired target victim because he shows interest in finding this website. In this case, the compromised server will provide malicious content to the user (④). The malicious response could directly be malicious web content or it redirects the user to malicious websites through multiple redirection hops (⑤). However, if the request originates from direct users (not via specific search engines), the attackers will not intent to expose the malicious content. This cloaking technique can make the attack very stealthy. In this case, the website will return non-malicious content (⑥).

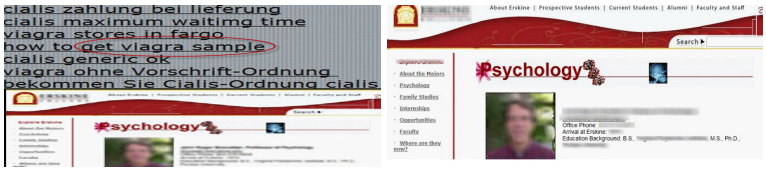
In our work, we define the web content responded by the compromised website (after redirection if it has) to the crawler bot as "Bot View", to users via the search engine as "Searcher View", and to users *not* via the search engine as "User View". We apply a similar technique used in [22, 25] to collect the ground truth on whether a website is compromised by search poisoning attack or not, i.e., whether its Searcher View and User View are different. More precisely, we *conservatively* consider the two views (Searcher/User) are different only when the final domain names (after redirection if there is any) are different [22]. In this way we can reduce false positives (due to dynamics in normal websites) and increase our confidence.[3]

2.2 Methods of Responding Malicious Content

As described in Section 2.1, in such search poisoning attacks, the compromised websites need to recognize the crawler bot to promote malicious content in search engines. Next, we describe two typical methods utilized by attackers to promote malicious content: tampering normal web pages, and creating new malicious web pages.

Tampering Normal Web Pages In this way, when attackers compromise the website, they will inject malicious content into normal web pages. Once the compromised website recognizes that a request is from a search crawler bot, it will reply with both injected malicious content and normal web page content. Once the compromised website recognizes that a request is from a user's direct visit (not referred from search engines), it will reply with the normal webpage. As a case study illustrated in Figure 2, an attacker compromised a

[3] Note that we may have very few false negatives using this conservative comparison. However that is not a problem for us because our goal is not on the precise detection but on the high efficiency in finding more compromised websites.

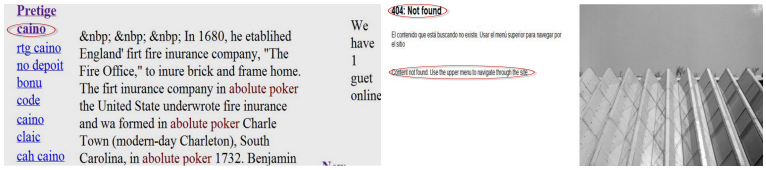


(a) Bot View (b) User View

Fig. 2. A case study of tampering normal web pages.

professor's personal homepage to launch search poisoning attack. Under such an attack, the Bot View contains both illicit pharmacy content such as "get Viagra sample" (as seen in the upper part of Figure 2(a)) and the professor's personal information (as seen in the lower part of Figure 2(a)), and the User View only contains correct personal information (as seen in Figure 2(b)).

Creating Malicious Web Pages In this way, unlike tampering existing normal web pages, attackers will upload or create totally new malicious web pages and only provide malicious content as Bot View to the search crawler bot. This content may be totally irrelevant to the themes of the whole website. As a case study illustrated in Figure 3, the attacker compromised a furniture company's website, which is implemented using a vulnerable version of WordPress [17]. Through exploiting the vulnerabilities of the WordPress, the attacker promoted casino content in Bot View to the search engine through creating a new malicious webpage hosted in the compromised website (as seen in Figure 3(a)). However, the attacker will provide a web page displaying "Not Founded" to users, who visit the same URL without using the search engine (as seen in Figure 3(b)).



(a) Bot View (b) User View

Fig. 3. A case study of creating malicious Web pages.

3 System Design

3.1 Intuition

Our design of PoisonAmplifier is based on the following three major intuitions:

Intuition 1: Attackers tend to use a similar set of keywords in multiple compromised websites (in the Bot View) to increase the visibility to desired users through search engines. Attackers usually artificially construct the content of Bot View, which will be indexed by search engines, to increase the chance of making compromised websites be searched through search engines. More specifically, similar to keyword stuffing [12], a common way of achieving this goal is to put popular keywords (those words are frequently searched by users on search engines such as Google Trends) into the Bot View. In this way, different compromised websites may share the similar popular keywords to draw attentions from victims. However, since many popular benign websites may also use these popular keywords and thus occupy high search ranks, it is difficult to guarantee high search ranks for

those compromised websites that may be not very popular. As a supplement, another way is to buy some "distinguishable keywords" from specific websites [11]. These keywords may be not so popular as those popular terms. However, they tend to have low competition in search engines, i.e., they are not widely contained in the websites and can be effectively used to search out target websites. Thus, through promoting these words in the Bot View, the compromised websites could occupy high search ranks when users query these keywords in search engines. Thus, attackers may use these "distinguishable keywords" in multiple compromised websites to increase their search ranks.

In addition, since some attackers desire to reply malicious content to their target victims rather than arbitrary users, they tend to put specific keywords into the Bot View of compromised websites, which have close semantic meanings to the promoted websites. For example, some attackers tend to post pharmacy words into the Bot View, because they will finally mislead victims who are interested in buying pharmacy to malicious websites selling illicit pharmacy. In this way, different attackers who promote similar malicious content may spontaneously use similar keywords in the Bot View.

Based on this intuition, once we obtain those specific keywords injected by attackers into the Bot View of known compromised websites, we can search these keywords in search engines to find more compromised websites.

Intuition 2: Attackers tend to insert links of compromised websites in the Bot View to promote other compromised websites; and the websites containing URLs linking to known compromised websites may also contain URLs linking to other unknown compromised websites. To increase the chance of leading victims to malicious websites, attackers usually use multiple compromised websites to deliver malicious content. Thus, to increase the page ranks of those compromised websites to search engines or to help newly created webpages on compromised websites to be indexed by search engines, attackers tend to link these compromised websites with each other by inserting links of compromised websites into the Bot View. In addition, attackers with different malicious goals may spontaneously promote links of compromised websites into the same popular third-party websites such as forums and online social network websites, either because these third-party websites are easy to be indexed by search engines or they do not have sanitation mechanisms. *Based on this intuition, we can find more compromised websites by searching the URLs in the Bot View linking to known compromised websites, and by searching the URLs in the web content of other websites, which have already been exploited to post URLs linking to known compromised websites.*

Intuition 3: Attackers tend to compromise multiple websites by exploiting similar vulnerabilities. Once attackers compromise some specific websites by exploiting their system/software vulnerabilities to launch search poisoning attacks, they tend to use similar tricks or tools to compromise other websites with similar vulnerabilities to launch search poisoning attacks. *Based on this intuition, once we know the vulnerabilities exploited by attackers to some compromised websites, we can find more compromised websites by searching websites with similar vulnerabilities.*

3.2 System Overview

We next introduce the system overview of PoisonAmplifier, based on the three intuitions described in Section 3.1. As illustrated in Figure 4, PoisonAmplifier mainly contains five components: Seed Collector, Promoted Content Extractor, Term Amplifier, Link Amplifier, and Vulnerability Amplifier.

- Similar to other existing work [27, 23], the goal of **Seed Collector** is to collect a seed set of compromised websites by searching initial key terms (e.g., Google Trends) in popular search engines.
- For each compromised website, **Promoted Content Extractor** will first work as a search engine bot to crawl the website's Bot View, and then work as a real user to obtain the websites' User View. Then, Promoted Content Extractor will extract those content that exists in the website's Bot View but does *not* exist in the website's User View. This content, defined as "promoted content", is essentially what attackers desire to promote into search engines.
- After extracting the promoted content, **Term Amplifier** extracts special key terms by analyzing the promoted content and querying these key terms in search engines to find more compromised websites.
- **Link Amplifier** extracts URLs in the promoted content. Link Amplifier will also extract URLs contained in the web content of third-party websites, which have already been posted links to known compromised websites. Then, Link Amplifier will analyze these URLs to find more compromised websites.
- By analyzing system/software vulnerabilities of those seed compromised websites and newly found compromised websites through using Term Amplifier and Link Amplifier, **Vulnerability Amplifier** finds more compromised websites by searching other websites with similar vulnerabilities.

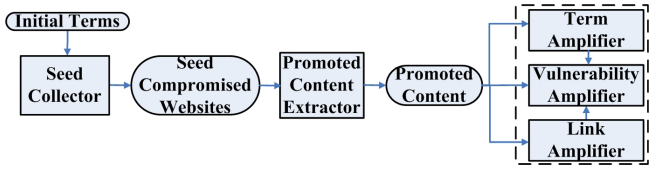


Fig. 4. The system architecture of PoisonAmplifier.

3.3 Seed Collector

As illustrated in Figure 5, Seed Collector mainly uses the following four steps to collect seed compromised websites: (1) it first uses Google Trends [6], Twitter trends[16], and our customized key terms as initial key terms to search on search engines. (2) For each term, it will extract the links of the top M search results showed in the search engine[4]. (3) For each link, Seed Collector crawls its Searcher View and User View through utilizing HttpClient-3.x package[10][5] to set different HTTP header parameters and values. Specifically, to crawl the Searcher View of the website linked by each search result, we send HTTP requests with customized Http Referrer (http://www.google.com/?q="term") to simulate a user to visit the website through searching Google. To crawl the User View, we send HTTP requests with customized values of UserAgent in the HTTP header (e.g., UserAgent: Mozilla/5.0 (Windows NT 6.1), AppleWebKit/535.2 (KHTML, like Gecko), Chrome/15.0.874.121, Safari/535.2) to simulate a user to directly visit the website. For both User View and Searcher View, the seed

[4] In our experiment, we choose $M = 200$.

[5] This package can handle HTTP 3xx redirection and provide flexible HTTP header configuration functions

collector follows their redirection chains and gets their final destination URL. (4) For each link, if its final destination domains between User View and Searcher View are different, we consider that this linking website is compromised and output it as a compromised website.

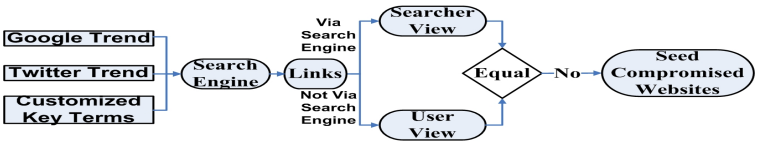


Fig. 5. Flow of Seed Collector.

3.4 Promoted Content Extractor

As described in Section 2.1, the essence of the search poisoning attack is to recognize different request origins and provide different web content to crawler bots (Bot View), to users via search engines (Searcher View), and to users not via search engines (User View). And attackers tend to inject specific content into the Bot View to increase the chances of their compromised websites to be searched out in search engines. They may also tend to inject malicious content that is related to the final promoted destination malicious websites. This content is usually different from normal web content, and can not be seen by users without using search engines.

The goal of the Promoted Content Extractor is to extract that injected content in the Bot View of known compromised webpages, which may also be contained in other compromised websites. Note that the Bot View may also contain normal content that is not injected by attackers and will be displayed in the User View. To be more effective, PoisonAmplifier only extracts and analyzes the content that is in the Bot View but is *not* in the User View, i.e., the content will be indexed by crawler bots, but not be seen by users directly visiting the websites. As illustrated in Figure 6, for each compromised website, Promoted Content Extractor crawls its Bot View and User View through sending crafted requests from crawler bots and users without using search engines, respectively. Specifically, to crawl the Bot View, we send request with customized value of UserAgent in the HTTP header (e.g., UserAgent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)) to mimic a Google bot visit. Promoted Content Extractor crawls the User View in the same way as Seed Collector. Then, Promoted Content Extractor extracts HTML content that appears in the Bot View but not in the User View. Then, it will further filter web content that is used for displaying in the web browsers such as HTML Tags and CSS codes, and also remove dynamic web function related codes such as Javascripts, which are not unique enough to help further amplification. Finally, it outputs this extracted "Promoted Content" after filtering.

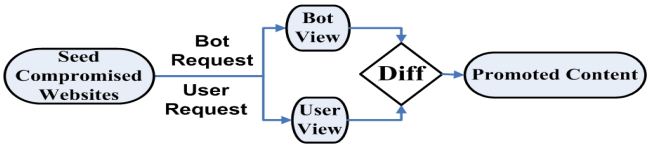


Fig. 6. Flow of Promoted Content Extractor.

It is worth noting that some legitimate websites with dynamic server-side codes can also return different content or even redirect to different websites for every request no matter where the visit is from (User View or Bot View), which may lead to false positives in our extracted promoted content. To decrease this kind of false positives, we crawl the User View twice within a short time period. In this case, if the two User Views are different, we will conservatively consider that this website is *not* compromised (even its User View and Searcher View may be different) and discard it for promoted content extraction.

3.5 Term Amplifier

Based on Intuition 1 in Section 3.1, the goal of Term Amplifier is to find more compromised websites through searching specific query terms extracted from promoted content.

It is worth noting that if we use less distinguishable content as query terms to search, we can obtain a higher recall number (more compromised websites could be returned) but a lower accuracy (top search results are less likely to be compromised websites), and vice versa. In addition, in order to obtain a higher accuracy, it is practical to focus on analyzing replied search results with top search ranks rather than analyzing all search results. Thus, the essential part of Term Amplifier is how to extract effective query terms from promoted content, through searching which we can obtain as many compromised websites as possible with a high accuracy. One option is to use each word/phrase in the content as one query term. However, in this way, some terms may be so general that most returned websites are benign, leading to a low accuracy. Another option is to use the "n-gram" algorithm [13] ($n \geq 2$). In this way, some terms may be so distinguishable that many compromised websites will be missed, leading to a low recall number.

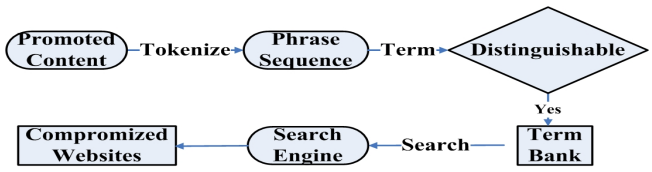


Fig. 7. Flow of Term Amplifier.

In our work, we design an algorithm, named "distinguishable n-gram", to extract query terms. As illustrated in Figure 7, Term Amplifier first tokenizes the promoted content into a sequence of phrases $\{P_i | i = 1, 2, \ldots, N\}$ by using the tokenizer of any non-Alphanumeric character except "blank", such as "comma", "semicolon", and "question mark". Then, for each phrase P_i, Term Amplifier will *exactly search* it on the search engine. If the number of returned search results SN_i is lower than a threshold T_D[6], we consider P_i as a "distinguishable" term and directly add it into a term set, named TermBank. Otherwise, Term Amplifier combines the phrases of P_i and P_{i+1} as a new query term to search. If this new term is "distinguishable", we add it into TermBank; otherwise, Term Amplifier combines the phrases of P_i, P_{i+1} and P_{i+2} as a new term to search. This process will continue until the number of phrase in the new term is equal to n. If the new term with n phrases is still not "distinguishable", the algorithm will discard the phrase P_i. In this way, TermBank comprises all the distinguishable terms

[6] T_D can be tuned with the consideration of the tradeoff between the accuracy and the recall number. In our preliminary experiment, we choose $T_D = 1,000,000$.

extracted from the promoted content. The detailed description of "distinguishable n-gram" algorithm is shown in Algorithm 1.

Algorithm 1 : Distinguishable n-gram Algorithm

```
Tokenize promoted content into phrases {P_i|i = 1, 2, ..., N}
for i := 1 to N do
  for j := 0 to n − 1 do
    Search "P_i P_{i+1} ... P_{i+j}" on the search engine to get SN_i
    if SN_i ≤ T_D then
      Add 'P_i P_{i+1} ... P_{i+j}" into TermBank
      CONTINUE
    end if
  end for
end for
Return TermBank
```

After building TermBank, similar to Seed Collector, Term Amplifier uses each query term in TermBank to search in the search engine and identifies compromised webpages through comparing their Searcher Views and User Views.

3.6 Link Amplifier

Based on Intuition 2 in Section 3.1, Link Amplifier first extracts two types of links: inner-links and outer-links. Inner-links refer to those links/URLs in the promoted web content of the compromised websites (as illustrated in the left part of Figure 8). Outer-links refer to those links/URLs in the web content of third-party websites, which have been posted with URLs linking to known compromised websites (as illustrated in the right part of Figure 8). We utilize Google dork [7] to locate the outer-links. For example, if one compromised website "seed.com" is obtained through searching one seed term "seedTerm", then we obtain those websites through searching "*intext:seed.com intext:seedTerm*" on Google. Then we crawl all the websites in search results, which usually are blogs or comments that contain "seed.com" and other scam links. Then, similar to Term Amplifier, for each inner-link and outer-link, Link Amplifier crawls its Searcher View and User View, and considers the linking website as compromised website if the Searcher View and User View are different.

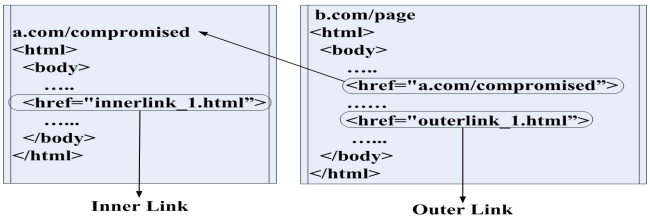


Fig. 8. The illustration of inner-links and outer-links.

We acknowledge that since those third-party websites may also post many benign links, many of outer-links will not link to compromised websites, leading to a relatively low accuracy. However, one of the benefits is that, through analyzing those outer-links, we can find more categories of compromised websites. For example, through analyzing outer-links from

compromised websites selling illicit pharmacy, we could find compromised websites that promote other topics such as "adult/sexy content". One case study of a forum webpage posting outer-links to both "adult" and "pharmacy" websites can be seen in Appendix A. Furthermore, we can still somehow increase the accuracy through focusing on only those third-party websites that have posted scam terms. This is because that this kind of websites are more likely to be used to promote malicious content by attackers than other websites. Thus, the links posted in such websites are more suspicious.

3.7 Vulnerability Amplifier

Once an attacker compromises a website to launch search poisoning attack by exploiting specific system/software vulnerabilities of the websites, it is very likely that he uses the same vulnerability to compromise more websites. For example, once some attackers know about the vulnerabilities of some specific version of "WordPress" [17] and successfully use some existing tools to compromise some websites that are implemented through using that specific version of WordPress, they may try to find other vulnerable websites that are also implemented with that version of WordPress. One possible simple way of finding those vulnerable websites could be to search keywords such as "powered by WordPress" on search engines.

Based on Intuition 3 in Section 3.1, Vulnerability Amplifier essentially mimics the way of attackers to find those compromised websites. Specifically, Vulnerability Amplifier first collects compromised websites by using Term Amplifier and Link Amplifier. Then, it will analyze possible system/software vulnerabilities of those compromised websites and extract the web content signature of the websites that utilize those vulnerable software. In our preliminary work, we only focus on analyzing the vulnerabilities of one specific software WordPress[7], which is a very popular target for attackers[1]. For example, one vulnerability of "Timthumb.php" in the WordPress themes allows attackers to upload and execute arbitrary php scripts. Vulnerability Amplifier will find compromised websites through searching those websites that use WordPress and contain at least one scam word. Since the URLs of the websites developed using WordPress typically contain a string of "wp-content", we can find those websites through searching Google Dork "*inurl:wp-content intext:scamWord*". After visiting each of such websites, Vulnerability Amplifier examines whether it is compromised or not by comparing its Searcher View and User View.

Currently, Vulnerability Amplifier still requires some manual work to extract search signatures. In the future, we plan to incorporate some techniques similar to existing automatic signature generation studies (e.g., AutoRE [30]) to automate some of the tasks.

4 Evaluation

In this section, we evaluate PosionAmplifier in two stages. For the first stage, we evaluate PoisonAmplifier regarding its effectiveness, efficiency, and accuracy with first 7 days' data. We also check the "discovery diversity" among different components in terms of finding exclusive compromised websites, i.e, how different the discovered compromised websites by different components are. In addition, we examine how existing Google security diagnostic tools in labeling malicious/compromised websites work on our found compromised websites. In the second stage, we extend the time to 1 month to verify if the PoisonAmplifier can constantly find new compromised websites.

[7] Even though we only analyze the vulnerabilities of one specific software in this work, our approach can easily include other types of system/software vulnerabilities, which is our future work.

4.1 Evaluation Dataset

As mentioned in Section 3, the seed term set consists of three categories: Google Trends, Twitter Trends and our customized keywords. For the Google Trend Topics, we crawled 20 Google Trend keywords each day for a week. In this way, we collected 103 unique Google Trends topics. For the Twitter Trends, we collected top 10 hottest Twitter trends each day for a week. In this way, we collected 64 unique Twitter Trends topics. For the customized key terms, we chose one specific category of scam words – pharmacy words[8]. Specifically, we chose 5 pharmacy words from one existing work [26], which provides several categories of scam words. We also manually selected another 13 pharmacy words from several pharmacy websites. Table 1 lists all 18 pharmacy words used in our study.

Table 1. 18 seed pharmacy words

kamagra	diflucan	levitra	phentermine	propecia	lasix
viagra	amoxil	xanax	cialis	flagyl	propeciatramadol
zithromax	clomid	Viagra super active	cialis super active	cipro	pharmacy without prescription

Then, for each of 18 pharmacy words, we obtained another 9 Google Suggest words through Google Suggest API [8]. In this way, we finally collected 165 unique pharmacy words. Table 2 summarizes the number and unique number of seed terms for each category.

Table 2. The number of seed terms for three different categories

Category	**# of terms**	**# of unique terms**
Google Trend	140	103
Twitter Trend	70	64
Pharmacy	180	165
Total	390	332

Then, for each of these 332 unique seed terms, we searched it on "Google.com" and collected the top 200 search results[9]. Then, for each search result, we use the similar strategy as in [22] to determine whether a website is compromised by examining whether the domain of its Searcher View and User View are different. In this way, we finally obtained 252 unique seed compromised websites through using those 332 seed terms. We denote this dataset as S_I, which is used in Stage I. After one week's amplification process, we denote the amplified terms and compromised websites from Stage I as S_{II}, which is the input for Stage II to recursively run PoisonAmplifier for 1 month.

4.2 Evaluation Results

Effectiveness. To evaluate the effectiveness of our approach, we essentially check how many new compromised websites can be found through amplifying dataset S_I. To measure the

[8] In our preliminary experiment, we only use pharmacy words. However, our approach is also applicable to other categories of words such as "adult words" or "casino words".

[9] In our experiment, we only focus on the search poisoning attacks on Google. However, our approach can be similarly extended to other search engines such as "yahoo.com" and "baidu.com". Also, the number of 200 can be tuned according to different experiment settings.

effectiveness, we use a metric, named "Amplifying Rate (AR)", which is the ratio of the number of newly found compromised websites to the number of seed compromised websites. Thus, a higher value of AR implies that the approach is more effective in finding compromised websites based on the seed compromised websites.

Table 3 shows the number of newly found compromised websites for each component. We can see that Term Amplifier has the highest AR of 323, which confirms that Term Amplifier can be very effective in discovering compromised websites. Even though Inner-link Amplifier and Outer-link Amplifier have relatively lower ARs than Term Amplifier, they can still discover over 10 times more compromised websites from the seeds. Actually, the reason why Term Amplifier can obtain a higher AR is mainly because we can extract much more query terms than inner-links and outer-links from the promoted content. In this way, we can essentially search out much more websites that contain the query terms from the search engine. In addition, even though we only focus on analyzing one specific software in our Vulnerability Amplifier, we can still discover over 4 times more compromised websites from the seeds. Overall, starting from only 252 seed compromised websites, these four strategies can totally discover around 75,000 unique compromised websites, and achieve a overall high amplifying rate of 296. The distribution information of these compromised websites in terms of their Top Level Domain(TLD) is show in Figure 10(a).

Table 3. The effectivenss of PoisonAmplifier.

Component	# seed compromised website	# unique compromised websites	Amplifying Rate
TermAmplifier	252	69,684	323.03
Inner-linkAmplifier	252	2,468	10.63
Outer-linkAmplifier	252	2,401	10.34
VulnerabilityAmplifier	252	482	4.49
Total (Unique)	252	74,671	296.31

Efficiency. To evaluate the efficiency of our approach, we essentially examine whether the websites visited by PoisonAmplifier are more likely to be compromised websites or not. To measure the efficiency, we use another metric, named "Hit Rate (HR)", which is the number of newly found compromised websites to the total number of websites visited by PoisonAmplifier. Thus, a higher Hit Rate implies that our approach is more efficient, because it means our approach can find more compromised websites by visiting fewer websites. Next, we evaluate the efficiency of individual amplification component, as well as the efficiency of different types of query keywords.

Component Efficiency. Table 4 shows the number of visited websites, the number of newly found compromised websites, and the values of hit rate for each component.

Table 4. The efficiency of different components.

Component	# Visited Websites	# Compromised Websites	Hit Rate
TermAmplifier	684,540	69,684	10.18%
Inner-linkAmplifier	3,097	2,468	79.69%
Outer-linkAmplifier	353,475	2,401	0.68%
VulnerabilityAmplifier	45,348	482	1.06%
Total	1,086,496	74,671	6.87%

From this table, we can see that Inner-link Amplifier can achieve the highest hit rate of 79.69%. This confirms that attackers tend to promote links of compromised websites to the search engine bots. The hit rate of Term Amplifier is around 10%, which is lower than that of Inner-link Amplifier. However, Term Amplifier can discover much more compromised websites than that of Inner-link Amplifier in terms of overall quantity, because we essentially extract significantly more terms than inner-links to search on the search engine. The hit rate of Outer-link Amplifier is relatively low, which is mainly because most of those outer-links are benign or do not have redirections. However, through using Outer-link Amplifier, we can find new types of scam terms promoted by different attackers. This is very useful to increase the diversity of the seed terms and to find more types of compromised websites. Vulnerability Amplifier also has a relatively low hit rate, because most top ranked websites with "WordPress" are benign. However, similar to Outer-link Amplifier, Vulnerability Amplifier also provides a method to find more (new) types of scam words and compromised websites.

Term Efficiency. We also analyze the term efficiency in finding compromised websites, i.e., which kinds of terms can be used to efficiently search out compromised (rather than normal) websites. Specifically, we compare three types of terms: seed terms (those 332 seed terms used in the Seed Collector), promoted phrases (the sequence of phrases obtained through tokenizing promoted content), and distinguishable terms (all the terms in TermBank obtained by utilizing "Distinguishable n-gram Algorithm"). Essentially, we use these three types of terms to search on Google to find compromised websites by utilizing Term Amplifier.

As seen in Figure 9, among these three types of terms, our extracted distinguishable terms can achieve the highest hit rate. Specifically, around 60% of distinguishable terms' hit rates are less than 0.2, while around 80% of promote phrases and 90% of seed terms have such values. This implies that using distinguishable terms is more effective to find compromised websites. In addition, over 60% of seed terms' hit rates are nearly zero, which shows that the current pre-selected terms are not very efficient compared to our new terms extracted from promoted content.

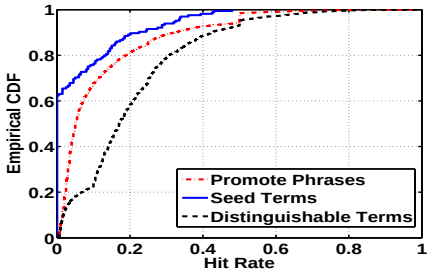


Fig. 9. Hit rate distribution.

To find what specific terms are most efficient, we further analyze the terms with the top five hit rates in TermBank. As seen in Table 5, we can see that all these five terms' hit rates are higher than 79%. In addition, we also find that three of these five terms have the same semantic meanings of sub-phrase as "No Prescription Needed". That may be due to the reason that attackers frequently use such phrases to allure victims, because many kinds of pharmacy drugs need prescription to buy in reality. The other two terms contain the names of two popular drugs: "Diflucan" (an anti-fungal medicine) and "Nimetazepam" (working specifically on the central nervous system).

Table 5. Terms with Top Five Hit Rates

Term	Hit Rate
Order Diflucan from United States pharmacy	90%=180/200
Buy Online No Prescription Needed	87%=174/200
Buy Cheap Lexapro Online No Prescription	85%=170/200
Online buy Atenolol without a prescription	83%=166/200
Nimetazepam interactions	79.5%=159/200

Diversity among different components. In this section, we analyze the diversity of newly found compromised websites among different components, i.e., we examine how many compromised websites of each component are exclusive, which can not be found by other components. The intuition is that if one component can find the compromised websites that can not be found by another component, then these components are very complementary and they can be combined together to be effective in discovering compromised websites. To measure the diversity, we use a metric, named "Exclusive Ratio (ER)", which is the ratio of the number of compromised websites that are only found by this component to the total number of compromised websites found by this component.

As seen in Table 4, we can find that all four components can obtain high exclusive ratios, higher than 88%. This observation shows that all these four components are complementary and it makes perfect sense to combine them together to achieve high effectiveness in discovering new compromised websites. Also, we can find that Term Amplifier's exclusive ratio is over 99%. That is mainly because Term Amplifier can find more compromised websites through visiting more websites.

Table 6. Exclusive ratio of different components

Component	TermAmplifier	Inner-linkAmplifier	Outer-linkAmplifier	VulnerabilityAmplifier
Exclusive Ratio	99.56%	96.11%	89.09%	88.77%

Comparison with existing work. In this experiment, we first compare the performance of PoisonAmplifier with two existing work: Leontiadis et al. [22] and Lu et al. [23] (both of them use pre-selected terms). To further verify the performance of the pre-selected term method used in above two work, we tested this method with our dataset. Table 7 shows the comparison result.

Table 7. The comparison of effectiveness with existing work

Research Work	# Seed Terms	# Visited Websites	# Compromised Websites	Hit Rate	Time
Leontiadis et al. [22]	218	3,767,040	63,000	1.67%	9 months
Lu et al. [23]	140	500,000	1,084	0.2%	1 months
Pre-selected terms	322	64,400	252	0.39%	7 days
PoisonAmplifier	332	1,086,496	74,671	6.87%	7 days

From this table, we can see that compared with [22], based on a similar number of seed terms, our work can find more compromised websites with a higher hit rate within a significantly shorter period. Also, compared with [23], our approach uses much fewer seed terms, but

discovers much more compromised websites with a much higher hit rate within a significantly shorter period. Compared with pre-selected terms method, with the same number of seed terms and same evaluation time, our approach can find much more compromised websites. Also, the hit rate of our approach is the highest, which is around 4 times and 34 times as that of [22] and [23], respectively. This observation shows that our approach is more efficient and effective in discovering/collecting compromised websites, since our approach does not highly rely on the pre-selective keywords (pre-selective keywords typically lead to a low hit rate, which has also been verified by [19]), which are used by both existing approaches.

Comparison with Google security diagnostic tools. We conducted another experiment to further evaluate the effectiveness of our approach. We want to test whether our newly found compromised websites are also detected in a timely fashion by Google's two state-of-the-art security diagnostic tools: Google Safe Browsing (GSB) [5] and Google Security Alert [4]. GSB is a widely used URL blacklist to label phishing websites and malware infected websites. Google Security Alert is another security tool, which labels compromised websites through showing the message "This site maybe compromised" within Google search results.

We first check how many new compromised websites found by each component are labelled as "phishing" or "malware infected". As seen in Table 8, we found that GSB labels only 547 websites as "malware infected" and zero as "phishing" through examining all $74,671$ newly found compromised websites. We next check how Google Security Alert works on

Table 8. Labeling results by using GSB

Component	# Compromised Websites	# Phishing	# Malware Infected
TermAmplifier	69,684	0	536
Inner-linkAmplifier	2,468	0	2
Outer-linkAmplifier	2,401	0	3
VulnerabilityAmplifier	482	0	6
Total (Unique)	74,671	0	547

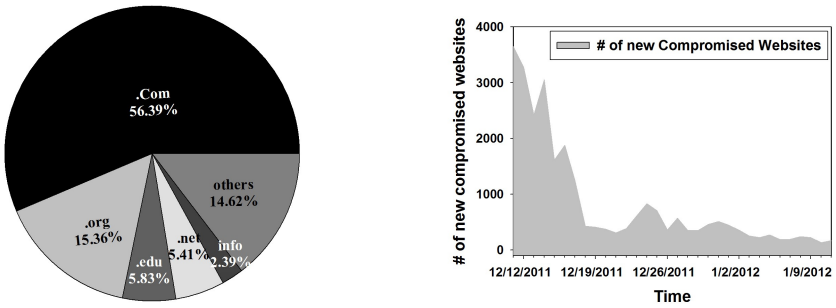
our newly found compromised websites. Specifically, we sampled 500 websites (which were randomly selected from those $74,671$ compromised websites) and finally found none of them were labelled as compromised.

Through the above experiments, we can find that most of our newly found compromised websites have not been detected by current Google security diagnostic tools. Although we do not argue that our approach is more effective than those two Google security tools, this observation shows that our approach can be effectively utilized to discover many new compromised websites that Google has not yet detected.

Accuracy. In this paper, we collect the ground truth through comparing the difference between Searcher View and User View, which proves to be a conservative and effective approach to identify search poisoning attacks [22, 25]. To further gain more confidence, we have conducted a manual verification on 600 randomly sampled URLs from all labelled compromised websites, and all of these sample websites were manually verified as indeed compromised websites.

Constancy. To evaluate the constancy of our approach, we essentially examine whether PoisonAmplifier can continue to find new compromised websites over time. Figure 10(b) is the distribution of new crawled compromised websites in Stage II. We can see that during the first several days, our system can find more new compromised websites because Term Amplifier inherits a large number of terms from data S_I. With these terms, our system can

efficiently find other compromised websites sharing similar terms. After that, the daily newly found compromised websites decrease quickly due to the exhaustion of terms. However, Link Amplifier and Vulnerability Amplifier can keep finding new terms and compromised websites everyday because the attackers keep promoting and attacking everyday. In this case, our system can still constantly find new compromised websites everyday leading to 26,483 new found compromised websites during 1 month's recursive amplification process.



(a) Distribution based on Top Level Domain (b) Daily new found compromised websites.

Fig. 10. Statistics of found compromised websites.

5 Limitations

In this section, we will discuss the limitations of our work.

We first acknowledge that since we mainly utilize pharmacy keywords as initial terms in our evaluation, this method may generate some bias. We use illicit pharmacy as a specific case study to evaluate our approach mainly because it is a typical target of search poisoning attack. However, our approach can be easily applied to other scenarios such as fake AntiVirus or web scams through changing customized keywords. In addition, through our evaluation, we can also observe that even though we use pharmacy keywords as initial customized keywords, those newly found compromised websites could be injected with content related to other scenarios. Thus, PoisonAmplifier can discover a broader range of compromised websites, instead of being restricted to only those used to promote illicit pharmacy by attackers.

We also acknowledge that since it is still difficult for Promoted Content Extractor to accurately filter all dynamic content, this may decrease the performance (in terms of hit rate) of our approach. However, visiting websites multiple times can somehow relieve this kind of problem. In addition, we indeed manually checked several hundred randomly sampled compromised websites and we have not found such kind of false positives so far. Also, our Distinguishable n-gram Algorithm can filter some general terms (generate by dynamic content) and reduce their impact.

In addition, we realize that once attackers know about the principle of our approach, they may try to evade our approach through providing non-malicious content to our Bot View with the utilization of IP-based cloaking techniques. For example, they may refuse to deliver malicious content if they find the IP address from our crafted Google bot crawler does not match known addresses of Google. However, as an alternative technique of Bot View by manipulating Http Referer, we can use the cache results of search engines such as Google cache as Bot View. In such way, we can obtain the Bot View of those compromised websites, as long

as attackers want to make their content crawled and indexed by popular search engines to launch search poisoning attacks. Besides, attackers may also try to decrease the effectiveness of our approach through inserting noisy content into their injected content. However, if the noisy content is general, our system will drop them based on our "Distinguishable n-gram Algorithm". Otherwise we can still consider these noisy data as "real" promoted content as long as they are shared in multiple compromised websites.

6 Related Work

Measurement and Understanding of Search Poisoning Attacks. Cloaking techniques are commonly used in search poisoning attacks, which have been analyzed and measured in many existing work [27] [28] [29]. Wu et al. present the earliest measurement study of cloaking techniques [27]. This work provides a way of identifying cloaking techniques through crawling webpages multiple times (using both user and crawler identifiers) and comparing the replied content.

In terms of search poisoning attacks, Wang et al. [25] investigate the dynamic property and show that the majority of compromised websites remain high in search results. Leontiadis et al. present a detailed measurement study of search poisoning attacks in the scenario of illicit online pharmacy [22]. Moore et al. provide an in-depth study on Google Trending Terms and Twitter Trending Topics that are abused to attract victims [24]. Through analyzing 60 million search results and tweets, this work characterizes how trending terms are used to perform search poisoning attacks and to spread social network spam.

Detection of Search Poisoning Attacks. Besides understanding search poisoning attacks, several approaches have been proposed to detect such attacks. John et al. [21] analyze a specific case study of search poisoning attacks and propose an automatic detection method based on detection insights obtained through the observation that the new created page(named SEO page in the paper) always contains trending terms and exhibit patterns not previously seen by search engines on the same domain. Lu et al. [23] present an in-depth study on analyzing search poisoning attacks and redirection chains, then build a detection system based on several detection features extracted from browser behaviors, network traffic, and search results.

Unlike most existing studies that try to understand or detect search poisoning attacks, our work focuses more on efficiently and effectively identifying(amplifying) more websites compromised by the search poisoning attacks, given a small seed set. We think this is an important problem not addressed so far. Our work is essentially motivated by these existing studies and is complementary to them.

In addition, the intuition behind our work is that we try to use attackers' tricks against them. Specifically, our work tries to find compromised websites through exploiting attackers' promoted content, which are injected by the attackers to attract the search engine bot and search traffic. In such case, John et al. [20] have similar ideas but target on a different problem, in which the authors propose a framework to find more malicious queries by generating regular expressions from a small set of malicious queries. In a recent concurrent study, EvilSeed[19] also shares similar inspiration but with different target and techniques. It searches the web for pages that are likely malicious by starting from a small set of malicious pages. To locate the other nearby malicious pages, they design several gadgets to automatically generate search queries. However, with the three oracles used in their work, Google's Safe Browing blacklist[5], Wepawet[18], and a custom-built tool to detect sites that host fake AV tools, EvilSeed cannot handle more stealthy attacks such as Search Poisoning Attacks discussed in this paper. That is,

EvilSeed can only find a small subset of these cloaking attacks that PoisonAmplifier can find. In addition, since PoisonAmplifier extracts the content that the attackers intend to promote while EvilSeed uses much more generic signatures in its SEO gadget, PoisonAmpifier can find more search poisoning compromised websites more *efficiently* and *effectively* than EvilSeed, e.g., the hit rate of EvilSeed is 0.93% in its SEO gadget compared with 6.87% hit rate in PoisonAmplifier. We consider PoisonAmplifier as a good complement to EvilSeed.

7 Conclusion

In this work, we have designed and implemented a novel system, PoisonAmplifier, to discover compromised websites that are utilized by attackers to launch search poisoning attack. Based on intrinsic properties of search poisoning attack, PoisonAmplifier first extracts attackers' promotion content in a small seed set of known compromised websites. Then, PoisonAmplifier utilizes Term Amplifier and Link Amplifier to find more compromised websites through searching specific terms and links in those promotion content on search engines. PoisonAmplifier also utilizes Vulnerability Amplifier to find more compromised websites, which have similar system/software vulnerabilities to existing known compromised websites. Our evaluation shows that PoisonAmplifier can find nearly 75,000 compromised websites by starting from 252 verified compromised websites within first 7 days. Also, compared with two related work, PoisonAmplifier can find around 4 times and 34 times compromised websites by analyzing the same number of websites.

8 Acknowledgment

This material is based upon work supported in part by the National Science Foundation under Grant CNS-0954096 and the Texas Higher Education Coordinating Board under NHARP Grant no. 01909. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and the Texas Higher Education Coordinating Board.

References

1. 50,000 websites infected with spam from 'wplinksforwork'. http://news.softpedia.com/news/50-000-Websites-Infected-with-Spam-From-Wplinksforwork-223004.shtml/.
2. Cloaking. http://en.wikipedia.org/wiki/Cloaking.
3. Cmu researcher finds web hackers profiting from illegal online pharmacies. http://www.darkreading.com/insider-threat/167801100/security/client-security/231400204/cmu-researcher-finds-web-hackers-profiting-from-illegal-online-pharmacies.html.
4. Google fights poisoned search results. http://www.securitynewsdaily.com/google-poisoned-search-results-0603/.
5. Google safe browsing. http://code.google.com/apis/safebrowsing/.
6. Google trend. http://www.google.com/trends.
7. Googledork. http://googledork.com/.
8. Googlesuggest. http://code.google.com/p/google-refine/wiki/SuggestApi.
9. Hiding text with css for seo. http://www.seostandards.org/seo-best-practices/hiding-text-with-css-for-seo.html.
10. Httpclient. http://hc.apache.org/httpclient-3.x/.
11. The keyword shop. http://www.blackhatworld.com/blackhat-seo/buy-sell-trade/.
12. Keyword stuffing. http://www.seo.com/blog/keyword-stuffing/.
13. N-gram algorithm. http://en.wikipedia.org/wiki/N-gram.

14. The pharmacy example. `http://www.cmu.edu/news/stories/archives/2011/august/aug11_onlinepharmacyhackers.html`.
15. Royal wedding, obama birth certificate search poisoned with fake av links. `http://www.eweek.com/c/a/Security/Royal-Wedding-Obama-Birth-Certificate-Search-Poisoned-with-Fake-AV-Links-489242/`.
16. Trending topics. `http://support.twitter.com/entries/101125-about-trending-topics`.
17. Word press. `http://wordpress.com/`.
18. M. Cova, C. Kruegel, and G. Vigna. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code. In *In International World Wide Web Conference (WWW'10)*, 2010.
19. L. Invernizzi, P. Comparetti, Stefano Benvenuti, C. Kruegel, M. Cova, and G. Vigna. EVILSEED: A Guided Approach to Finding Malicious Web Pages. In *In IEEE Symposium on Security and Privacy (Oakland).*, 2012.
20. J. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy. Searching the Searchers with SearchAudit. In *In Proceedings of the 19th USENIX Security*, 2010.
21. J. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy. deSEO: Combating search-result poisoning. In *In Proceedings of the 20th USENIX Security*, 2011.
22. N. Leontiadis, T. Moore, and N. Christin. Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade. In *In Proceedings of the 20th USENIX Security*, 2011.
23. L. Lu, R. Perdisci, and W. Lee. SURF: Detecting and Measuring Search Poisoning. In *In Proceedings of ACM Conference on Computer and Communications Security (CCS'11)*, 2011.
24. T. Moore, N. Leontiadis, and N. Christin. Fashion Crimes: Trending-Term Exploitation on the Web. In *In Proceedings of ACM Conference on Computer and Communications Security (CCS'11)*, 2011.
25. D. Wang, S. Savage, and G. Voelker. Cloak and Dagger: Dynamics of Web Search Cloaking. In *In Proceedings of ACM Conference on Computer and Communications Security (CCS'11)*, 2011.
26. Y. Wang, M. Ma, Y. Niu, and H. Chen. Double-Funnel: Connecting Web Spammers with Advertisers. In *In proceedings of the 16th international conference on World Wide Web, pages 291C300, 2007*, 2007.
27. B. Wu and B. Davison. Cloaking and redirection: A preliminary study. In *In Adversarial Information Retrieval on the Web(AIRWeb)*, 2005.
28. B. Wu and B. Davison. Identifying link farm spam pages. In *In Special Interest Tracks and Posters of the International Conference on World Wide Web*, 2005.
29. B. Wu and B. Davison. Detecting semantic cloaking on the Web. In *In Proceedings of International Conference on World Wide Web (WWW'06)*, 2006.
30. Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov. Spamming Botnet: Signatures and Characteristics. In *In Proceedings of ACM SIGCOMM 2008*, 2008.

A Case Study of Outer-link

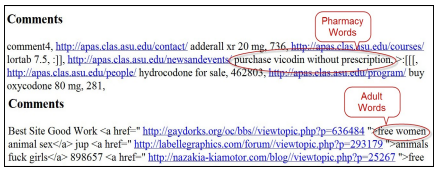


Fig. 11. Example of outer-link.

As seen in Figure 11, we first find the forum webpages through searching the websites that are known compromised websites - here is a pharmacy target compromised website. Then, through analyzing the forum webpage's content, we can also find other compromised websites with "Adult" content.