Take This Personally: Pollution Attacks on Personalized Services

Xinyu Xing, Wei Meng, Dan Doozan, Alex C. Snoeren[†], Nick Feamster, and Wenke Lee
Georgia Institute of Technology and [†]UC San Diego

Abstract

Modern Web services routinely personalize content to appeal to the specific interests, viewpoints, and contexts of individual users. Ideally, personalization allows sites to highlight information uniquely relevant to each of their users, thereby increasing user satisfaction—and, eventually, the service's bottom line. Unfortunately, as we demonstrate in this paper, the personalization mechanisms currently employed by popular services have not been hardened against attack. We show that third parties can manipulate them to increase the visibility of arbitrary content—whether it be a new YouTube video, an unpopular product on Amazon, or a low-ranking website in Google search returns. In particular, we demonstrate that attackers can inject information into users' profiles on these services, thereby perturbing the results of the services' personalization algorithms. While the details of our exploits are tailored to each service, the general approach is likely to apply quite broadly. By demonstrating the attack against three popular Web services, we highlight a new class of vulnerability that allows an attacker to affect a user's experience with a service, unbeknownst to the user or the service provider.

1 Introduction

The economics of the Web ecosystem are all about clicks and eyeballs. The business model of many Web services depends on advertisement: they charge for prime screen real estate, and focus a great deal of effort on developing mechanisms that make sure that the information displayed most prominently is likely to create revenue for the service, either through a direct ad purchase, commission, or at the very least improving the user's experience. Not surprisingly, malfeasants and upstanding business operators alike have long sought to reverse engineer and exploit these mechanisms to cheaply and effectively place their own content—whether it be items for sale, malicious content, or affiliate marketing schemes. Search engine optimization (SEO), which seeks to impact the placement of individual Web pages in the results provided by search engines, is perhaps the most widely understood example of this practice.

Modern Web services are increasingly relying upon personalization to improve the quality of their customers' experience. For example, popular websites tailor their front pages based on a user's previous browsing history at the site; video-sharing websites such as YouTube recommend related videos based upon a user's watch history; shopping portals like Amazon make suggestions based on a user's previous purchases; and search engines such as Google return customized results based upon a wide variety of user-specific factors. As the Web becomes increasingly personal, the effectiveness of broad-brush techniques like SEO will wane. In its place will rise a new class of schemes and outright attacks that exploit the mechanisms and algorithms underlying this personalization. In other words, personalization represents a new attack surface for all those seeking to steer user eyeballs, regardless of their intents.

In this paper, we demonstrate that contemporary personalization mechanisms are vulnerable to exploit. In particular, we show that YouTube, Amazon, and Google are all vulnerable to the same class of cross-site scripting attack, which we call a *pollution attack*, that allows third parties to alter the customized content the services return to users who have visited a page containing the exploit. Although the attack is quite effective, we do not claim that it is the most powerful, broadly applicable, or hard to defeat. Rather, we present it as a first example of a class of attacks that we believe will soon—if they are not already—be launched against the relatively unprotected underbelly of personalization services.

Our attack exploits the fact that a service employing personalization incorporates a user's past history (including, for example, browsing, searching and purchasing activities) to customize the content that it presents to the

user. Importantly, many services with personalized content log their users' Web activities whenever they are logged in regardless of the site they are currently visiting; other services track user activities on the site even if the user is logged out (*e.g.*, through a session cookie). We use both mechanisms to pollute users' service profiles, thereby impacting the customized content returned to the users in predictable ways. Given the increasing portfolio of services provided by major players like Google and Amazon, it seems reasonable to expect that a large fraction of users will either be directly using the service or at least logged in while browsing elsewhere on the Web.

We show that pollution attacks can be extremely effective on three popular platforms: YouTube, Google, and Amazon. A distinguishing feature of our attack is that it does not exploit any vulnerability in the user's Web browser. Rather, it leverages these services' own personalization mechanisms to alter user's experiences. While our implementation employs cross-site request forgery (XSRF) [13], other mechanisms are possible as well.

The ability to trivially launch such an attack is especially worrisome because it indicates the current approach to Web security is ill-equipped to address the vulnerabilities likely to exist in personalization mechanisms. In particular, today's Web browsers prevent exploits like cross-site scripting and request forging by enforcing boundaries between domains though "same origin" policies. The limitations of these approaches are well known, but our attack represents a class of exploits that cannot be stopped by client-side enforcement: in an attempt to increase the footprint of its personalization engine (*e.g.*, Google recording search queries that a user enters on a third-party page), a service with personalized services is providing the cross-site vector itself. Hence, only the service can defend itself from such attacks on its personalization. Moreover, enforcing isolation between independent Web sessions seems antithetical to the goal of personalization, which seeks to increase the amount of information upon which to base customization attempts.

This paper makes the following contributions:

- We describe pollution attacks against three platforms—YouTube, Google, and Amazon—that allow a third party to alter the personalized content these services present to users who previously visited a Web page containing the exploit.
- We study the effectiveness of our attack on each of these platforms and demonstrate that it (1) can increase the visibility of almost any YouTube channel; (2) dramatically increase the ranking of most websites in the short term, and even have lasting impacts on the personalized rankings of a smaller set of sites, and (3) cause Amazon to recommend reasonably popular products of the attacker's choosing.
- Our attack and its effectiveness illustrates the importance of securing personalization mechanisms in general. We discuss a number of implications of our study and ways for websites to mitigate similar vulnerabilities in the future.

The rest of the paper is organized as follows. Section 2 provides a general overview of pollution attacks on personalized services. Sections 3, 4, and 5 introduce specific attacks that can be launched against YouTube, Google, and Amazon, respectively, and report on our success. We survey related work in Section 6 and discuss limitations of our work and possible defenses in Section 7 before concluding in Section 8.

2 Overview and Attack Model

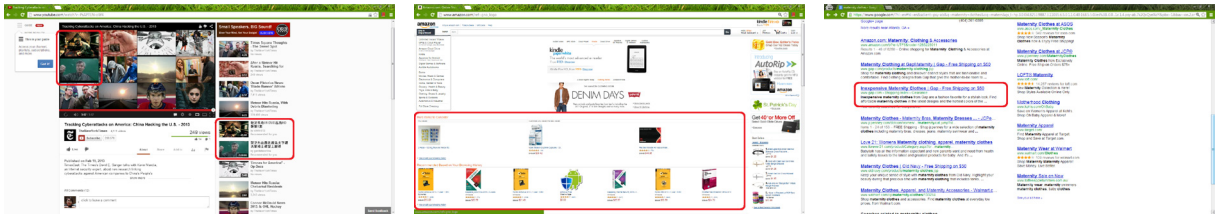
In this section, we present a brief overview of personalization as it is used by popular Web services. We then present a model of pollution attacks, which we apply to three different scenarios later in the paper: YouTube, Amazon, and Google.

2.1 Personalization

Online services are increasingly using personalization to deliver information to users that is tailored to their interests and preferences. Personalization potentially creates a situation where both the service provider and the user benefit: the user sees content that more closely matches preferences, and the service provider presents products that the user is more likely to purchase (or links that the user is more likely to click on), thus potentially resulting in higher revenues for the service provider.

The main instrument that a service provider can use to affect the content that a user sees is modifying the *choice set*, the set of results that a user sees on a particular screen in response to a particular query. The size of a choice set differs for different services. For example, YouTube shows the user anywhere from 12–40 videos; Amazon may show the user up to five sets of recommended products; Google's initial search results page shows the top ten results. Figure 1 shows several examples of choice sets on different sites.

When a user issues a query, a service's *personalization algorithm* affects the user's choice set for that query. The choice set that a personalization algorithm produces depends on a user query, as well as a number of auxiliary factors, including the universe of all possible content and the user's browsing history. Previous work has claimed that many factors, ranging from geography to time of day, may affect the choice set that a user sees. For the purposes of the attacks in this paper, we focus on how changes to a user's history can affect the choice set,



(a) Customized YouTube. (b) Customized Amazon. (c) Customized Google.

Figure 1: websites with personalized services (personalized services tailor the data in the red rectangles).

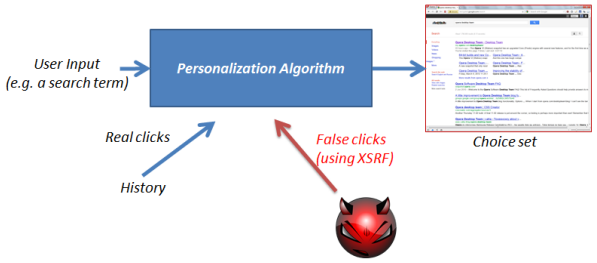


Figure 2: Overview of how history pollution can ultimately affect the user's choice set.

holding other factors fixed. In particular, we study how an attacker can pollute the user's history by generating false clicks through cross-site request forgery (XSRF). We describe these attacks in the next section.

2.2 Pollution Attacks

The objective of a pollution attack is to affect a user's *choice set*, given a particular input. In some cases, a user's choice set appears before the user enters any input (*e.g.*, upon an initial visit to the page). In this case, the attacker's goal may be to affect a default choice set. Figure 2 shows an overview of the attacker's goal: the attacker aims to affect the resulting choice set by altering the user's history with false clicks, using cross-site request forgery as the attack vector. This attack requires three steps:

1. *Model the service's personalization algorithm.* We assume that the attacker has some ability to model the personalization algorithm that the site uses to affect the user's choice set. In particular, the attacker must have some idea of how the user's past history affects the user's choice set. This information is often available in published white papers, but in some cases it may require experimentation.
2. *Create a "seed" to pollute the user's history.* Given some knowledge of the personalization algorithm and a goal for how to affect the choice set, the attacker must design the seed that is used to affect the user's choice set. Depending on the service, the seed may be queries, clicks, purchases, or any other activity that might go into the user's history. A good seed can affect the user's choice set with a minimal number of "false clicks", as we describe next.
3. *Inject the seed with a vector of false clicks.* To pollute a user's history, in most cases we require that the user be signed in to the site. (For some services, pollution can take place even when the user is not signed in.) Then, the attacker can use a mechanism to make it appear as though the user is taking action on the Web site for a particular service (*e.g.*, clicking on links) using a particular attack vector.

In the following sections, we explore how an attacker can apply this same procedure to attack the personalization algorithms of three different services: YouTube, Amazon, and Google search.

3 Pollution Attacks on YouTube

In this section, we demonstrate our attack on YouTube[1]. Following the attack steps we described in Section 2, we first model how YouTube uses the watch history of a YouTube user account to recommend videos by reviewing the literature [5]. Second, we discuss how to prepare seed data (*i.e.*, seed videos) to promote target data (*i.e.*, target videos belonging to a specific channel). Third, we introduce how to inject the seed videos to a YouTube user account. Finally, we design experiments and quantify the effectiveness of our attack.

3.1 YouTube Personalization

YouTube constructs a personalized list of recommended videos based upon the videos a user has previously viewed [5]. YouTube attempts to identify the subset of previously viewed videos that the user enjoyed by considering only those videos that the user watched for a long period of time. Typically, YouTube recommends videos that other users with similar viewing histories

[1] A demo video is available at `http://www.youtube.com/watch?v=8hij52ws98A`.

have also enjoyed. YouTube tracks the co-visitation relationship between pairs of videos, which reflects how likely a user who watched a substantial portion of video X will also watch and enjoy video Y. In general, there may be more videos with co-visitation relationships than there is display area, so YouTube prioritizes videos with high rankings. YouTube will not recommend a video the user has already watched.

YouTube displays recommended videos in the suggestion list placed alongside with a playing video (*e.g.*, Figure 5) and in the main portion of the screen at the end of a video (Figure 1(a)). A suggestion list appearing next to a video typically contains 20–40 suggested videos, two of which are recommended based upon personalization. At the end of a video, YouTube shows an more concise version of the suggestion list that contains only twelve of the videos from the full list; these videos may or may not contain personal recommendations.

3.2 Preparing Seed Videos

YouTube organizes videos into channels, where each channel corresponds to the set of uploads from a particular user. In our attack, we seek to promote a set of target videos, Ω^T, all belonging to the same YouTube channel, C. To do so, we will use an additional set of seed videos, Ω^S, that have a co-visitation relationship with the target videos. By polluting a user's watch history with videos in Ω^S, we can cause YouTube to recommend videos in Ω^T. There are two ways to obtain Ω^S: we can identify videos with pre-existing co-visitation relationships to the target videos, or we can create the relationships ourselves.

Existing Relationships. In the simplest version of the attack, the attacker identifies existing videos to use as the seed set. For example, given a target video set Ω^T belonging to channel C, the attacker could consider all of the other videos in the channel, $C - \Omega^T$, as candidate seeds. For every candidate video, the attacker checks which videos YouTube recommends when a fresh YouTube account (*i.e.*, a YouTube account with no history) watches it. YouTube allows its users to view their recommended videos at `http://www.youtube.com/feed/recommended`. If the candidate video triggers YouTube to recommend a video in Ω^T, then the attacker adds the injected video to seed video set Ω^S.

In general, this process allows the attacker to identify seed videos for every target video in Ω^T. The attacker cannot yet launch the attack, though, because a YouTube video in Ω^S may trigger YouTube to also recommend videos not in Ω^T. To address this issue, the attacker can simply add these unwanted videos to the seed video set Ω^S because YouTube does not recommend videos that the user has already watched. As we will show later, the attacker can convince YouTube that the user watched, but *did not* enjoy, these unwanted videos, so their inclusion in Ω^S will not lead to additional recommendations.

Fabricating Relationships. For some videos, it may be difficult to identify a seed set Ω^S that recommends all of the elements of Ω^T due to lack of co-visitation relationships for some of the target elements. Instead, attackers who upload their own content to use as the seed set can create co-visitation relationships between this content and the target set. In particular, an attacker uploads a set of videos, Ω^0, and establishes co-visitation relationships between Ω^0 and Ω^T through crowd-sourcing (*e.g.*, Mechanical Turk or a botnet): YouTube visitors need only watch a video in Ω^0 followed by a video in Ω^T. After a sufficient number of viewing pairs, the attacker can use videos in Ω^0 as the seed set. As we will show in Section 3.4.1, a relatively small number of viewing pairs suffices.

3.3 Injecting Seed Videos

To launch the attack and inject seed videos into a victim's YouTube watch history, an attacker can harness XSRF to forge the following two HTTP requests for each video in the seed set: (1) `http://www.youtube.com/user_watch?plid=<value>&video_id=<value>`, and (2) `http://www.youtube.com/set_awesome?plid=<value>&video_id=<value>`, where `plid` and `video_id` correspond to the values found in the source code of the seed video's YouTube page. The first HTTP request spoofs a request from the victim to start watching the seed video, and the second convinces YouTube that the victim watched the video for a long period of time. Both HTTP requests are required for videos in Ω^S to trigger the recommendation of videos in Ω^T, but only the first HTTP request is needed to prevent the recommendation of unwanted videos.

3.4 Experimental Design

We evaluated the effectiveness of our attack both in controlled environments and against real YouTube users. We first validated the the attack in the simplest scenario, where the attack promoted existing YouTube channels through existing co-visitation relationships. We then considered the scenario where an attack seemed to upload and promote content from a channel that the attacker created. Finally, we conducted a small-scale experiment to demonstrate the effectiveness of the attack against a volunteer set of real YouTube users.

3.4.1 New Accounts

We first promoted existing YouTube channels by launching our attack against victims with fresh YouTube user accounts. This experiment confirms the effectiveness of our approach in the absence of other, potentially countervailing influences, such as recommendations based on a user's existing history.

We began by selecting 100 existing YouTube channels at random from the list of the top 2,000 most-subscribed channels published by VidStatsX [19]. For each of the selected YouTube channels, we randomly selected 25 videos from the channel as the target video set, used the method described in the previous section to identify a seed video set, and injected the seed videos to a fresh YouTube account.

We then considered promoting new content by creating our own YouTube channel and similarly attacking fresh YouTube accounts. Our YouTube channel contains two 3-minute videos. We selected one of the videos as a one-element target video set and used the other as the seed set. We created a co-visitation relationship by embedding both videos on a web page and recruiting volunteers to watch both videos sequentially. We obtained 65 and 68 views for our seed and target video respectively.

3.4.2 Existing Accounts

We studied the effectiveness of our pollution attack using real YouTube user accounts. We recruited 22 volunteers with extensive pre-existing YouTube watch histories. To limit the inconvenience to our volunteers, we limited our study to attempting to promote one moderately popular YouTube channel based upon existing co-visitation relationships. We selected a moderately popular account because a popular channel may be recommended anyway (regardless of out attack); conversely, an entirely new channel requires a certain amount of effort to establish the co-visitation relationships as described above and we have limited volunteer resources.

Based on these parameters, we arbitrarily selected the channel *OnlyyouHappycamp*. We believe this selection is a reasonable candidate to be promoted using our attack for several reasons. First, compared to popular channels, most videos in *OnlyyouHappycamp* have low view counts (about 2,000 view counts per video on average) and the number of subscribers to the channel is a similarly modest 3,552. Both of these are easily achievable by an attacker at fairly low cost[2]. Second, most videos in *OnlyyouHappycamp* are 22 minutes long, which makes them suitable for promotion. As we will explain in Section 3.5.1, the length of a target video affects its likelihood for being recommended as a result of a co-visitation relationship with another video.

[2]According to the prices in underground markets such as `freelancer.com` and `fiverr.com`, 40,000 view counts and 10,000 subscribers cost $15 and $30 US dollars, respectively.

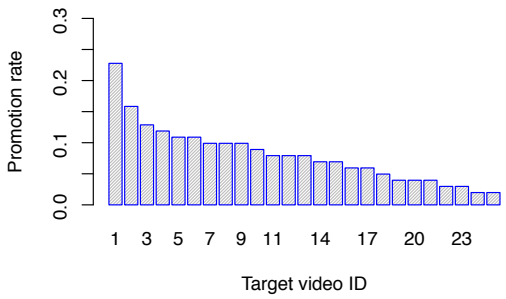


Figure 3: The promotion rate for each of the 25 target videos in channel *lady16makeup*. Two videos were recommended in each of the 114 trials.

Similar to the experiments with new accounts, we randomly selected 15 target videos from channel *OnlyyouHappycamp*, identified a seed set, and injected the seed videos into the volunteers' YouTube accounts. After pollution, the volunteers were asked to use their accounts to watch three videos of their choice and report the suggestion list displaying alongside each of their three videos.

3.5 Evaluation

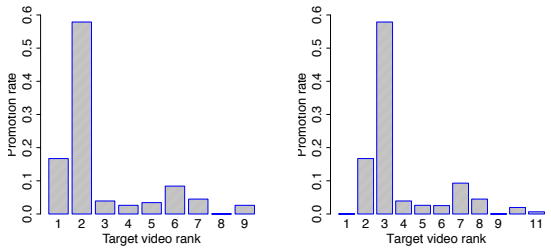
We evaluated the effectiveness of our pollution attacks by logging in as the victim user and viewing 114 representative videos[3]. We measured the effectiveness of our attack in terms of *promotion rate*: the fraction of the 114 viewings when at least one of the target videos was contained within the video suggestion list. Recall that the list contains at most two personalized recommendations (see Section 3.1); we deem the attack successful if one or both of these videos are videos that were promoted as a result of a pollution attack.

3.5.1 New Accounts

Pollution attacks successfully promoted target videos from each of the 100 selected existing channels: Each time we injected seed videos for a particular channel, we observed the target videos in the suggestion list for each of the 114 videos. Since these are fresh accounts, there is no other history, so our targeted videos always occupy both of the personalized recommendation slots.

In addition, we observed the particular target videos shown in the suggestion video list varied, even when

[3]We attempted to view 150 videos random from a trace of YouTube usage at our institution over the course of several months. Unfortunately, 36 of the videos were no longer available at the time of our experiment.



(a) Higher ranked video (b) Lower ranked video

Figure 4: Distribution of the suggestion slots occupied by each of the two successfully promoted target videos.

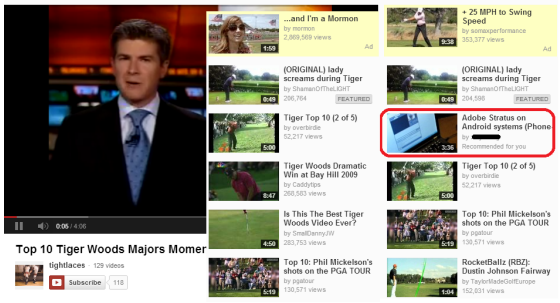


Figure 5: Suggestion lists before (left) and after (right) a pollution attack against a fresh YouTube user account. The video highlighted in red is our uploaded video.

we were viewing the same video using the same victim YouTube account. In other words, every target video has a chance to be promoted and shown on the suggestion video list no matter which video a victim plays. Figure 3 shows the frequency with which each of the 25 target videos for a representative channel, *lady16makeup*. In an attempt to explain this variation, we computed (1) the Pearson correlation between the showing frequencies and the lengths of the target videos for each channel (ρ_t); (2) the Pearson correlation between the showing frequencies and the view counts of these target videos for each channel (ρ_{cnt}). We found the average Pearson correlation values are medium ($\rho_t = 0.54$) and moderate ($\rho_{cnt} = 0.23$), respectively. This suggests that both the length and view count of a target video influence its recommendation frequency, but the length of a target video is a more significant factor.

Since screen real estate is precious, and users typically focus on the first few items of a list, we report on the position within the suggested video lists that our targeted videos occupied when they were promoted. We observed that the two target videos were usually placed back-to-back on the suggestion list. Figure 4 shows that YouTube usually placed our target videos among the top few spots of a victim's suggestion list: in our tests with new accounts, the target videos were always recommended and placed on the top 12, which meant they also appeared at the end of viewed videos. This finding is particularly significant because it implies that our target videos are shown even if a victim finishes watching a YouTube video on a third-party website (which typically embeds only the view-screen portion of the YouTube page, and not the full suggestion list).

Our attacks were similarly completely successful in promoting newly uploaded content. As a control, we also signed in as non-polluted fresh YouTube accounts and, unsurprisingly, did not find any of our new content among the videos in the suggestion list. In other words, the videos were recommended exclusively because of our attacks; our experiments were sufficiently

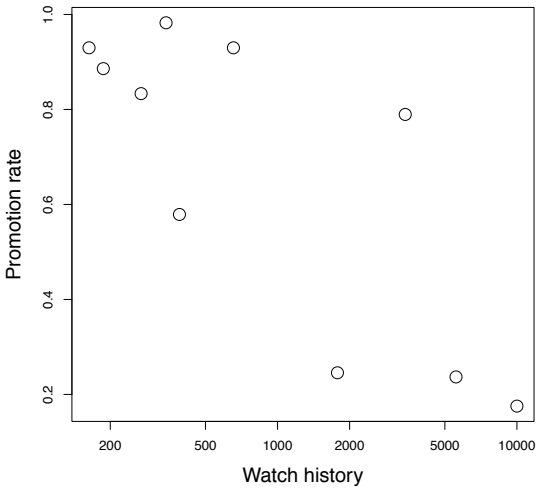


Figure 6: Promotion success rates for 10 real YouTube user accounts with varying watch history lengths.

small that we did not lead YouTube to conclude that our content was, in fact, universally popular. Figure 5 shows a sample screenshot comparing the suggestion lists from a victim account and another, non-exploited fresh account. Finally, we found that one of our target videos occupied the top suggestion slot while viewing 80 out of the 114 test videos.

3.5.2 Existing Accounts

Our attacks were somewhat less successful on real YouTube accounts. We found that 14 out of the 22 volunteer YouTube users reported that at least one of our target videos from channel *OnlyyouHappycamp* appeared in the suggestion list during each of their three video viewings, a 64% promotion rate.

To understand why we were able to exploit some accounts and not others, we asked our volunteers to share their YouTube watch histories. Ten of our volunteers shared their histories with us and allowed us to sign in to

their YouTube accounts to conduct a further study. The number of videos in the watch histories of the ten volunteers ranged from a few hundred to tens of thousands. Figure 6 shows the relationship between the number of watched videos in a watch history and the number of times that at least one of our target videos is displayed along with a playing video. While there appears to be an intuitive decreasing trend (i.e., the longer the history an account has the more resistant it is to pollution), there are obvious outliers. For example, one account with almost 3,500 previous viewings in its history succumbed to our attacks almost 80% of the time.

Consistent with the Pearson coefficients reported earlier, we found that the success of our attacks depends on the rankings and lengths of the videos that are otherwise suggested based upon a user's history. In particular, we observed that the majority of the videos recommended to users for whom our attacks have low promotion rates have longer lengths and more view counts than our target videos, while the videos that YouTube recommends based on the watch history of the user with 3,500 previous viewings have shorter lengths than our target videos (though they generally have higher view counts than our targets).

Although we believe our attack demonstrates that YouTube's personalization mechanism is subject to exploit, the persistence of the attack effects is unclear. In our experiments, volunteers watched arbitrary YouTube videos right after being attacked, but we believe our pollution attacks on YouTube are likely to last for some time. Although YouTube does not explicitly disclose how time factors into their recommendation system (if at all) [5], analysis of volunteers' watch histories indicates that a YouTube video that was watched as long as two weeks prior is still used for generating recommended videos.

4 Google Personalized Search

In this section, we show how history pollution attacks can be launched against Google's search engine[4]. The goal of our attack is to promote a target webpage's rank in the personalized results that Google returns for an arbitrary search term by injecting seed search terms into a victim's search history.

4.1 Search Personalization

Search personalization customizes search results using information about users, including their previous query terms, click-through data and previously visited websites. The details of Google's personalization algorithms are not public, but many previous studies have explored aspects of personalized search [2,4,6,7,9,10,14–18]. We describe two classes of personalization algorithms: *contextual personalization* and *persistent personalization*. According to recent reports [11,12], many search engines including Google, Bing, and Yahoo! apply both types of personalization.

Contextual personalization constructs a short-term user profile based on recent searches and clicks-through [4, 16]. When a user searches for "inexpensive furniture" followed by "maternity clothes," Google's contextual personalization algorithm typically promotes search results that relate to "inexpensive maternity clothes" for the next few searches (we provide an analysis of precisely how long this effect lasts in Appendix A.2). In contrast, persistent personalization uses the entire search history—as opposed to only recent searches—to develop a user profile [9, 15]. Personalization that occurs over the longer term may not affect a user's search results as dramatically, but can have longer-lasting effects for the results that a user sees. For example, searching for "Egypt" using different accounts may result in two distinct result sets: one about tourism in Egypt and one related to the Arab Spring.

4.2 Identifying Search Terms

Given the differing underlying algorithms that govern contextual and persistent personalization, an attacker needs to select different sets of seed search terms depending on the type of attack she hopes to launch.

Contextual Personalization. For the contextual personalization attack, the keywords injected into a user's search history should be both relevant to the promoting keyword and unique to the website being promoted. In particular, the keywords should be independent from other websites that have similar ranking in the search results, to ensure that only the target website is promoted. Presumably, an attacker promoting a specific website is familiar with the website and knows what keywords best meet these criteria, but good candidate keywords are also available in a website's `meta` keyword tag. While Google no longer incorporates `meta` tags into their ranking function [3], the keywords listed in the `meta` keyword tag still provide a good summary of the page's content.

Persistent Personalization. Launching a persistent personalization attack requires a different method of obtaining keywords to inject. In this case, the size of the keyword set should be larger than that used for a contextual attack in order to have a greater effect on the user's search history. Recall that contextual attacks only affect a user's current session, while persistent attacks pollute

[4] A demo video is available at `http://www.youtube.com/watch?v=73E5CLFYeu8`.

a user's search history in order to have a lasting effect on the user's search results. An attacker can determine suitable keywords using the Google AdWords tool, which takes as an input a search term and URL and produces a list of about one hundred related keywords. Ideally, an attacker could pollute a user's search history with each of these terms, but a more efficient attack should be effective with a much smaller set of keywords. We determined that an attacker can safely inject roughly 50 keywords a minute using cross-site request forgery; more rapid search queries are flagged by Google as a screen-scraping attack. For this study, we assume an attacker can inject at most 25 keywords into a user's profile, but the number of keywords can increase if the user stays on a webpage for more than 30 seconds. Not all keyword lists that AdWords returns actually promote the target website. The effectiveness of this attack likely depends on several factors, including the user's current search history. In Section 4.5, we evaluate the effectiveness of this attack under different conditions.

4.3 Injecting Search Terms

As with the pollution attacks on YouTube, the attack on Google's personalized search also uses XSRF to inject the seeds. For example, an attacker can forge a Google search by embedding `https://www.google.com/search?hl=en&site=&q=usenix+security+2013` into an invisible iframe. A Web browser will issue an embedded HTTP request, even if Google search response has an enabled `X-Frame-Option` header. Injecting search terms into a Google user's account affects the search results of the user's subsequent searches. The number and set of search terms to inject differs depending on whether an attacker can execute a contextual or persistent personalization attack.

4.4 Experimental Design

To cleanly study the effects of our proposed attacks on contextual and persistent search personalization, we conducted most of our experiments using Google accounts with no search history. To validate whether our results apply to real users, we also conducted a limited number of tests using accounts that we constructed to mimic the personae of real users.

To quantify the effectiveness of our attack in general, we must select an unbiased set of target web pages whose rankings we wish to improve. We built two test corpora, one for attacks on contextual personalization, and one for attacks on persistent personalization. We attempted to promote existing web sites using only their current content and link structure; we did not perform any SEO on websites before conducting the attacks. We believe this represents a conservative lower bound on the effectiveness of the attack, as any individual website owner could engineer the content of their site to tailor it for promotion through search history pollution.

4.4.1 Contextual Pollution

We started by scraping 5,671 shopping-related keywords from `made-in-china.com` to use as search terms. We then entered each of these terms into Google one-by-one to obtain the top 30 (un-personalized) search results for each. Since some of our search terms are related, not all of these URLs are unique. Additionally, we cannot hope to improve the URLs that are already top-ranked for each of the search terms. We obtained 151,363 URLs whose ranking we could hope to improve.

Because we cannot manually inspect each of these websites to determine appropriate seed search terms, we instead focused a subset that include the `meta` keyword tag. For the approximately 90,000 such sites, we extracted the `meta` keywords or phrases from the website. Many of these keywords are generic and will appear in a wide variety of websites. To launch the attack, we require keywords that are unique to the website we wish to promote (at least relative to the other URLs returned in response to the same query), so we ignored any keywords that were associated with multiple URLs in the same set of search results.

This procedure ultimately yielded 2,136 target URLs spanning 1,739 different search terms, for which we had a set of 1–3 seed keywords to try to launch a contextual pollution attack. The average search term has 1.23 results whose ranking we tried to improve. Figure 11 in the Appendix shows the distribution of the original rankings for each of these target websites; the distribution is skewed toward highly ranked sites, perhaps because these sites take care in selecting their `meta` tag keywords.

4.4.2 Persistent Pollution

Once again, we begin by selecting 551 shopping-related search terms and perform Google searches with each of the search terms to retrieve the top 30 search results. As opposed to the contextual attack, where we search for keywords that differentiate the results from one another, we aim to determine search terms that will be associated with the website and search-term pair for the long term.

As described in Section 4.2, we use a tool provided by Google AdWords to obtain a set of keywords that Google associates with the given URL and search term. Constructing related keyword lists for each of the 29 search returns (again excluding the top hit, which we cannot hope to improve) and 551 search terms yields 15,979 distinct URLs with associated lists of keywords.

For each URL, we select 25 random keywords from the AdWords list for 25 distinct trials. If a trial improved a URL's ranking, we then test the persistence of the attack by performing 20 subsequent queries, each with a randomly chosen set of Google trending keywords. These subsequent queries help us verify that the URL promotion is not just contextual, but does not vanish when a user searches other content. If after all 25 trials we find no keyword sets that promote the URL's ranking and keep it there for 20 subsequent searchers, we deem this URL attempt a failure. If multiple keyword sets succeed, we select the most effective (*i.e.*, the set of 25 keywords that induces the largest ranking improvement) trial to include in the test set.

4.5 Evaluation

In this section, we quantify the effectiveness of search history pollution with attacks that aimed to promote the target websites identified in the previous section. To scope our measurements, we consider the effectiveness of the attacks only for the set of search terms that we identify; it is quite possible, of course, that our pollution attacks also affect the rankings of the targeted URLs for other search terms.

When measuring the effectiveness of our attack, we use two different criteria, depending upon a website's original position in the search results. In the case of URLs that are already in the first ten search results but not ranked first, we consider the pollution attack successful if it increases the ranking of a URL at all. For URLs subsequent pages, we consider the attack successful only if the attack moves the URL to the first page of search results, since improved ranking on any page that is not the first page is unlikely to have any utility.

4.5.1 Top-Ranked Sites

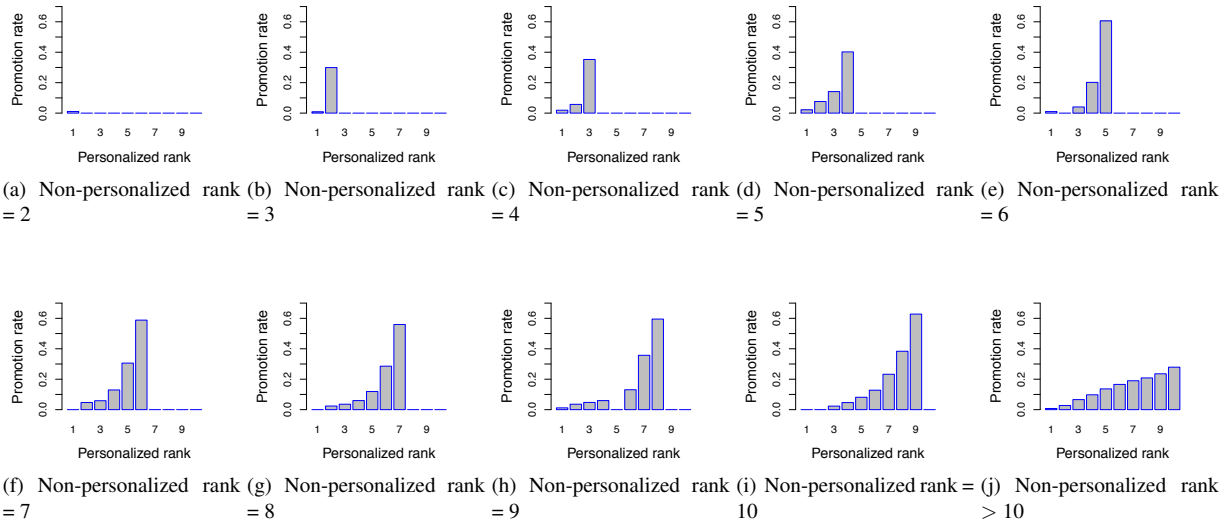
For the 2,136-page contextual attack test corpus, of the 846 pages that appeared on the front page prior to our attack, we improved the ranking of 371 (44%). The persistent attack was markedly less effective, with only 851 (17%) of the 4,959 test cases that originally appeared on the first page of the search results had ranking improvements surviving the persistence test (*i.e.*, they remained promoted after 20 random subsequent queries). In both cases, however, the probability of success depends greatly on the original ranking of the targeted URL. For example, promoting a second-ranked URL to the top-ranked position for contextual personalization succeeded 1.1% of the time, whereas promoting a tenth-ranked URL by at least one position succeeded 62.8% of the time. Similarly, for attacks on persistent personalization, moving a second-ranked URL to the top succeeded 4.3% of the time, and moving a tenth-ranked URL to a higher-ranked position succeeded 22.7% of the time. These results make sense, because second-ranked sites can only move into the top-ranked position, whereas sites that are ranked tenth can move into any one of nine higher spots.

To illustrate this effect and illuminate how far each webpage was promoted, Figure 7 shows the PDF of an improved webpage's rank after contextual history pollution, based upon its position in the non-personalized search results. We observed that contextual pollution was able to promote most webpages by one or two spots, but some low-ranking webpages were also promoted to very high ranks. Similarly, Figure 8 shows the distributions for each result ranking for those websites whose rankings were improved by a persistent history pollution attack. Here, the distributions appear roughly similar (although the absolute probability of success is much lower), but it is difficult to draw any strong conclusions due to the small number of promoted sites of each rank for either class of attack.

4.5.2 The Next Tier

The remaining 1,290 test websites for the contextual attack were initially on the second or third page of search results. By polluting a user's search history with the unique `meta` tag keywords associated with each site, we promoted 358 of them (28%) to the front page. Figure 7(j) shows that these websites were more likely to appear at the top of the results than those pages that were initially at the bottom of the first page. We suspect this phenomenon results from the choice of keywords used in pollution: because their original rankings were low, the pollution attack requires a distinguishing keyword to move one of the webpages to the front page at all. If such a keyword can move a search result to the first page, it might also be a good enough keyword to promote the page to a high rank on the first page, as well.

The results from the persistent test set are markedly different. Figure 8(j) shows that sites starting on the second or third page are unlikely to end up at the very top of the result list due to a persistent history attack: Only 80 (less than 1%) of the 11,020 attacks that attempted to promote a website appearing on the 2nd or 3rd page of results was successful in moving it to the front page (and keeping it there). This results shows that persistent search history attacks are generally best launched for sites that are already highly ranked, as opposed to contextual attacks, which can help even lower-ranked sites.



(a) Non-personalized rank = 2 (b) Non-personalized rank = 3 (c) Non-personalized rank = 4 (d) Non-personalized rank = 5 (e) Non-personalized rank = 6

(f) Non-personalized rank = 7 (g) Non-personalized rank = 8 (h) Non-personalized rank = 9 (i) Non-personalized rank = 10 (j) Non-personalized rank > 10

Figure 7: Promotion rates of promoted Google search rankings for successful contextual history pollution attacks.

4.5.3 Real Users

We also evaluate the effectiveness of pollution attacks on ten volunteers' accounts with extensive pre-existing search histories. We find that, on average, 97.1% of our 729 previously successful contextual attacks remain successful, while only 77.78% of the persistent pollution attacks that work on fresh accounts achieve similar success. We believe that users' search histories sometimes interfere with the attacks, and that user history interferes more with the attacks on persistent personalization. Contextualized attacks rely only on a small set of recent search terms to alter the personalized search results, which is unlikely to be affected by a user's search history. In contrast, pollution attacks against persistent personalization rely on more of a user's search history. If relevant keywords are already present in a user's search history, keyword pollution may be less effective. In any event, both attacks are relatively robust, even when launched against users with long search histories.

5 Pollution Attacks on Amazon

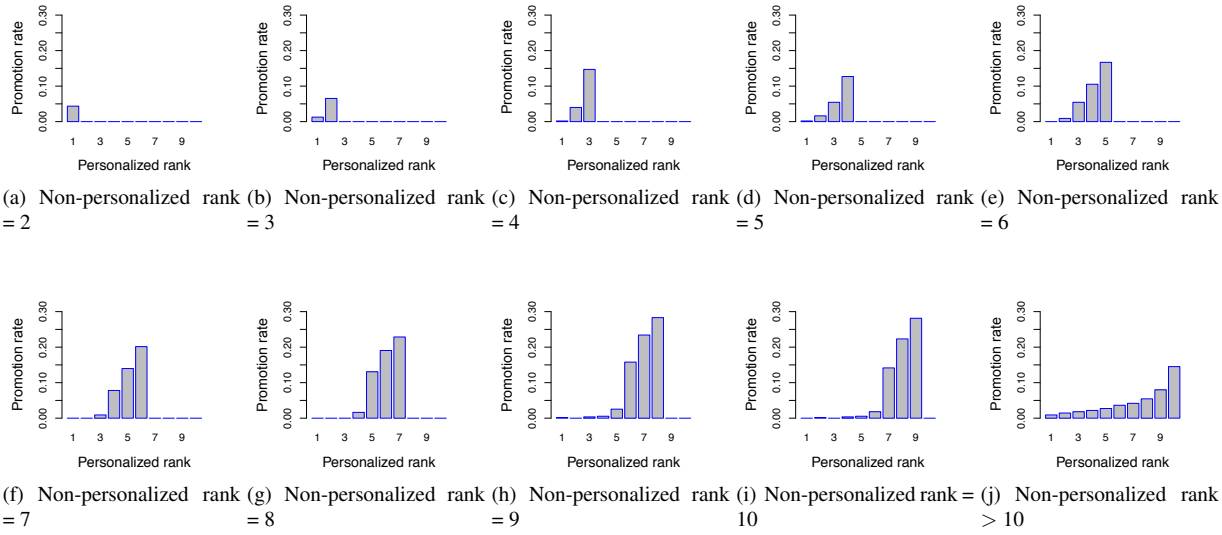
Of the three services, Amazon's personalization is perhaps the most evident to the end user. On one hand, this makes pollution-based attacks less insidious, as they will be visible to the observant user. On the other, of the three services, Amazon has the most direct monetization path, since users may directly purchase the goods from Amazon. Therefore, exploitation of Amazon's personalization may be profitable to an enterprising attacker.

Amazon tailors a customer's homepage based on the previous purchase, browsing and searching behavior of the user. Amazon product recommendations consider each of these three activities individually and explicitly labels its recommendations according to the aspect of the user's history it used to generate them. We focused on the personalized recommendations Amazon generates based on the browsing and searching activities of a customer because manipulating the previous purchase history of a customer may have unintended consequences.

5.1 Amazon Recommendations

Amazon displays five recommendation lists on a customer's homepage that are ostensibly computed based on the customer's searching and browsing history. Four of these lists are derived from the products that the customer has recently viewed (view-based recommendation); the fifth is based on the latest search term the customer entered (search-based recommendation). For each of the view-based recommendation lists, Amazon uses relationships between products that are purchased together to compute the corresponding recommended products; this concept is similar to the co-visitation relationship that YouTube uses to promote videos. For the recommendation list that is computed based on the latest search term of a customer, the recommended products are the top-ranked results for the latest search term.

In contrast to the types of personalization used for YouTube and Google Search, Amazon's personalization is based on history that maintained by the user's web browser, not by the service. Because customers frequently brows Amazon without being signed in, both the



(a) Non-personalized rank = 2 (b) Non-personalized rank = 3 (c) Non-personalized rank = 4 (d) Non-personalized rank = 5 (e) Non-personalized rank = 6

(f) Non-personalized rank = 7 (g) Non-personalized rank = 8 (h) Non-personalized rank = 9 (i) Non-personalized rank = 10 (j) Non-personalized rank > 10

Figure 8: Promotion rates of promoted Google search rankings for successful persistent history pollution attacks.

latest viewed products and search term of the customer are stored in session cookies on the user's browser rather than in profiles on Amazon servers.

5.2 Identifying Seed Products and Terms

Because Amazon computes the view and search-based recommendation lists separately, the seed data required exploit each list must also be different.

Visit-Based Pollution. To promote a targeted product in a view-based recommendation list, an attacker must identify a seed product as follows. Given a targeted product that an attacker wishes to promote, the attacker visits the Amazon page of the product and retrieves the related products that are shown on Amazon page of the targeted product. To test the suitability of these related products, the attacker can visit the Amazon page of that product and subsequently check the Amazon home page. If the targeted product appears in a recommendation list, the URL of the candidate related product can serve as a seed to promote the targeted product.

Search-Based Pollution. To promote a targeted product in a search-based recommendation list, it suffices to identify an appropriate search term. If automation is desired, an attacker could use a natural language toolkit to automatically extract a candidate keyword set from the targeted product's name. Any combination of these keywords that successfully isolates the targeted product can be used as the seed search term for promoting the targeted product. For example, to promote product "Breville BJE200XL Compact Juice Fountain 700-Watt Juice Extractor", an attacker can use XSRF to inject the search term "Breville BJE200XL" to replace an Amazon customer's latest search term.

5.3 Injecting Views and Searches

As with the attacks on the previous two services, the attacker embeds the Amazon URLs of the desired seed items or search queries into a website that the victim's browser is induced to visit with XSRF. For example, if one seed search terms is "Coffee Maker", the seed URL would be something like `http://www.amazon.com/s/?field-keywords=Coffee+Maker`. Similarly, an attacker could embed the URL of a seed product into an invisible *img* tag as the *src* of the image. When a victim visits the attacker's website, Amazon receives the request for that particular query or item and customizes the victim's Amazon website based on that search.

5.4 Experiment Design

To evaluate the effectiveness of the pollution attack against, we conducted two experiments. The first experiment measured the effectiveness of our attack when targeted toward popular items across different categories of Amazon products. The second quantified the effectiveness of our attack on randomly selected, mostly unpopular Amazon products.

5.4.1 Popular Products

Amazon categorizes sellers' products into 32 root categories. To select products from each category, we

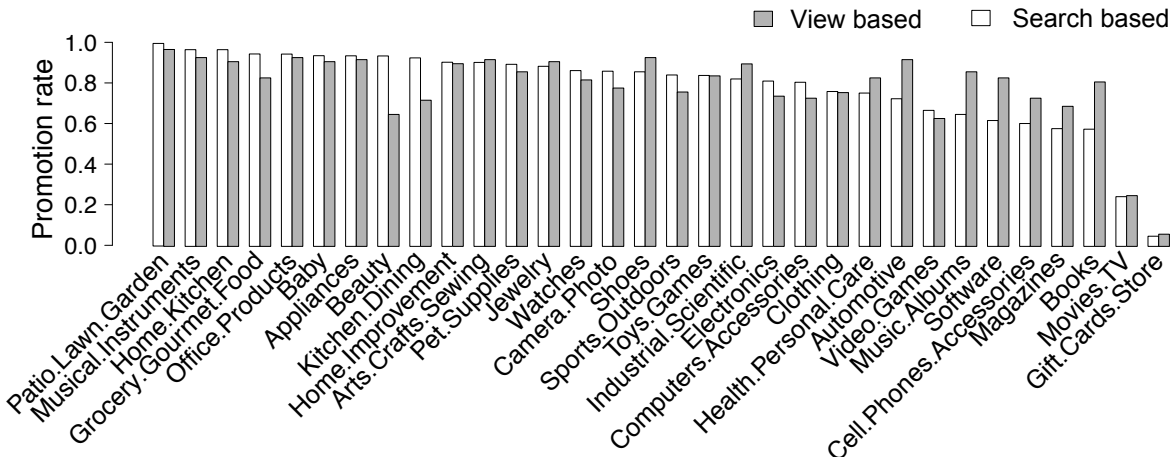


Figure 9: Promotion rates across Amazon categories.

scraped the top 100 best-selling products in each category in January 2013 and launched a separate attack targeting each of these 3,200 items.

5.4.2 Random Products

To evaluate the effectiveness of the polution attack for promoting arbitrary products, we also selected products randomly. We downloaded a list of Amazon Standard Identification Number (ASIN) [1] that includes 75,115,473 ASIN records. Because each ASIN represents a Amazon product, we randomly sampled ASINs from the list and constructed a set of 3,000 products currently available for sale. For every randomly selected product in the list, we recorded the sale ranking of that product in its corresponding category.

5.5 Evaluation

Because Amazon computes search and visit-based recommendations based entirely upon the most recent history, we can evaluate the effectiveness of the pollution attack without using Amazon accounts from real users. Thus, we measured the effectiveness of our attack by studying the success rate of promoting our targeted products for fresh Amazon accounts.

5.5.1 Promoting Products in Different Categories

To evaluate the effectiveness of the pollution attack for each targeted product, we checked whether the ASIN of the targeted product matches the ASIN of an item in the recommendation lists on the user's customized Amazon homepage.

Figure 9 illustrates the promotion rate of target products in each category. The view-based and search-based attacks produced similar promotion rates across all categories, about 78% on average. Two categories had significantly lower propotion rates: Gift-Cards-Store and Movies-TV (achieving 5% and 25%, respectively).

To understand why these categories yielded lower promotion rates, we analyzed the top 100 best selling products for each category. For Gift-Cards-Store, we found that there were two factors that distinguish gift cards from other product types. First, the gift cards all had similar names; therefore, using the keywords derived from the product name resulted in only a small number of specific gift cards being recommended. Second, we found that searching any combination of keywords extracted from the product names always caused a promotion of Amazon's own gift cards, which may imply that it is more difficult to promote product types that Amazon competes with directly.

Further investigation into the Movies-TV category revealed that Amazon recommends TV episodes differently. In our attempts to promote specific TV episodes, we found that Amazon recommends instead the first or latest episode of the corresponding TV series or the entire series. Because we declared a promotion successful only if the exact ASIN appears in the recommendation lists, these alternate recommendations are considered failures. These cases can also be considered successful because the attack caused the promotion of very similar products. Therefore, we believe that for all categories except for Gift-Cards-Store, an attacker has a significant chance of successfully promoting best-selling products.

5.5.2 Promoting Randomly Selected Products

We launched pollution attacks on 3,000 randomly selected products. We calculated the *Cumulative Success Rate* of products with respect to their rankings. The Cu-

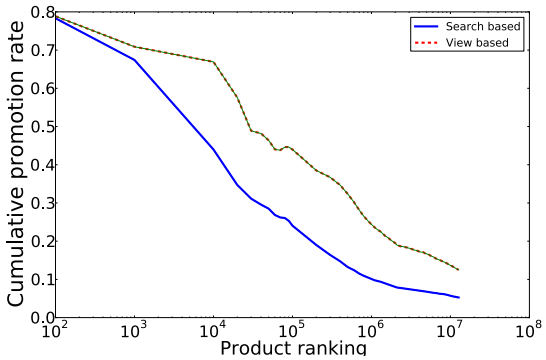


Figure 10: Cumulative promotion rates across varying product ranks for different Amazon pollution attacks.

mulative Success Rate for a given range of product rankings is defined as the ratio of the number of successfully promoted products to the number of target products in that range.

Figure 10 shows the cumulative promotion rate for different product rankings for the two different types of pollution attacks. As the target product decreases in popularity (*i.e.*, has a higher ranking position within its category) pollution attacks become less effective, but this phenomenon reflects a limitation of Amazon recommendation algorithms, not our attack. Products with low rankings might not be purchased as often; as a result, they may have few and weak co-visit and co-purchase relationships with other products. Our preliminary investigation finds that products which rank 2,000 or higher within their category have at least a 50% chance of being promoted by a visit-based pollution attack, and products with rankings 10,000 and higher have at least a 30% chance to be promoted using search-based attacks.

6 Related Work

To the best of our knowledge, the line of work most closely related to ours is black-hat search engine optimization (bSEO). Although sharing a common goal as search history pollution—illicitly promoting website rankings in search results—bSEO follows a completely different approach, exploiting a search engine's reliance on crawled Web content. Blackhat SEO engineers the content of and links to Web pages to obtain a favorable ranking for search terms of interest [8]. Thus, techniques that address bSEO are unlikely to be effective against pollution attacks. On the other hand, because bSEO targets the general indexing and ranking process inside search engines, any successfully promoted website will be visible to all search engine users, potentially significantly boosting the volume of incoming traffic. Yet, effective bSEO campaigns typically involve support from a complex network infrastructure, which may consist of hundreds of search-indexed websites (preferably with non-trivial reputations at established search engines) to coordinate and form a link farm [20]. These infrastructures not only require a considerable amount of money to build and maintain, but also take time to mature and reach their full effectiveness [8]. By contrast, launching a search history pollution attack is significantly easier.

We showed in Section 4 that a user's personalized search results can be manipulated simply by issuing crafted search queries to Google. Without requiring any external support, the entire process happens instantly while the user is visiting the offending Web page. Although our attack targets individual search users (*i.e.*, the polluted result is only visible to individual victims), it by no means limits the scale of the victim population, especially if an exploit is placed on a high-profile, frequently visited website.

7 Discussion

Our current study has several limitations. Most notably, the scale of our experiments is modest, but because we typically randomly select the target items, we believe that the results of our experiments are representative, and that they illustrate the substantial potential impacts of pollution attacks. Similarly, our specific pollution attacks are fragile, as each service can take relatively simple steps to defend againt them.

A possible defense against pollution attacks arises from the fact that cross-site request forgery can be stopped if requests to a website must carry tokens issued by the site. Enforcing this constraint, however, also prevents information and behaviors at third-party sites from being harvested for personalization and hampers the current trend of increasing the scope of data collection by websites for improved personalization. One short-term effect from this study may be that (some) websites will begin to consider the tradeoffs between the security and benefits of personalization.

YouTube in particular uses two separate HTTP requests to track a YouTube's user viewing activity that are independent from the act of streaming of the video. One straightforward defense against pollution attacks is to monitor the time between the arrivals of the two HTTP requests. If YouTube finds the interval is substantially less than the length of the video, it could ignore the signal. An attacker can still always inject a short video or control the timing of the HTTP requests in an effort to bypass such a defense mechanism. We did notice that an injected short video can be used to promote multiple longer videos; for example, watching a single two-

second video[5] causes YouTube to recommend several long videos.

8 Conclusion

In this paper, we present a new attack on personalized services that exploits the fact that personalized services use a user's past history to customize content that they present to the user. Our attack pollutes a user's history by using cross-site request forgery to stealthily inject and execute a set of targeted browsing activities in the user's browser, so that when the user subsequently accesses the associated service specific content is promoted. We illustrate how an attacker can pollute a user's history to promote certain content across three platforms. While our attack is simple, its impact can be significant if enough users' histories are compromised.

As personalization algorithms and mechanisms increasingly control our interactions with the Internet, it is inevitable that they will become the targets of financially motivated attacks. While we demonstrate pollution attacks on only YouTube, Google, and Amazon, we believe that our methods are general and can be widely applied to services that leverage personalization technologies, such as Facebook, Twitter, Netflix, Pandora, etc. The attacks we present here are just the first few examples of potentially many possible attacks on personalization. With increasingly complex algorithms and data collection mechanisms aiming for ever higher financial stakes, there are bound to be vulnerabilities that will be exploited by motivated attackers. The age of innocence for personalization is over; we must now face the challenge of securing it.

Acknowledgments

This research was supported in part by the National Science Foundation under grants CNS-1255453, CNS-1255314, CNS-1111723, and CNS-0831300, and the Office of Naval Research under grant 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 or the Office of Naval Research.

References

[1] Amazon.com product identifiers. `http://archive.org/details/asin_listing`.

[2] BENNETT, P. N., RADLINSKI, F., WHITE, R. W., AND YILMAZ, E. Inferring and using location metadata to personalize web search. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2011).

[3] CUTTS, M. Does Google use the "keywords" meta tag? `http://www.youtube.com/watch?v=jK7IPbnmvVU`.

[4] DAOUD, M., TAMINE-LECHANI, L., AND BOUGHANEM, M. A session based personalized search using an ontological user profile. In *Proceedings of The 24th Annual ACM Symposium on Applied Computing* (2009).

[5] DAVIDSON, J., LIEBALD, B., LIU, J., NANDY, P., VAN VLEET, T., GARGI, U., GUPTA, S., HE, Y., LAMBERT, M., LIVINGSTON, B., AND SAMPATH, D. The YouTube video recommendation system. In *Proceedings of the 4th ACM Conference on Recommender Systems* (2010).

[6] DOU, Z., SONG, R., AND WEN, J.-R. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th ACM International Conference on the World Wide Web* (2007).

[7] LIU, F., YU, C., AND MENG, W. Personalized web search by mapping user queries to categories. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management* (2002).

[8] LU, L., PERDISCI, R., AND LEE, W. Surf: detecting and measuring search poisoning. In *Proceedings of the 18th ACM Conference on Computer and communications security* (2011).

[9] MATTHIJS, N., AND RADLINSKI, F. Personalizing Web search using long term browsing history. In *The Fourth ACM International Conference on Web Search and Data Mining* (2011).

[10] QIU, F., AND CHO, J. Automatic identication of user interest for personalized search. In *Proceedings of the 15th ACM International Conference on the World Wide Web* (2006).

[11] SEARCH ENGINE LAND. Bing results get localized & personalized. `http://searchengineland.com/bing-results-get-localized-personalized-64284`.

[12] SEARCH ENGINE LAND. Google now personalizes everyones search results. `http://searchengineland.com/google-now-personalizes-everyones-search-results-31195`.

[13] SHIFLETT, C. Cross-site request forgeries. `http://shiflett.org/articles/cross-site-request-forgeries`, 2004.

[14] SIEG, A., MOBASHER, B., AND BURKE, R. Web search personalization with ontological user profiles. In *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management* (2007).

[15] SONTAG, D., COLLINS-THOMPSON, K., BENNETT, P. N., WHITE, R. W., DUMAIS, S., AND BILLERBECK, B. Probabilistic models for personalizing Web search. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining* (2012).

[5] `http://www.youtube.com/watch?v=UPXK3AeRvKE`

[16] SRIRAM, S., SHEN, X., AND ZHAI, C. A session-based search engine. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2004).

[17] TAN, C., GABRILOVICH, E., AND PANG, B. To each his own: personalized content selection based on text comprehensibility. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining* (2012).

[18] TEEVAN, J., DUMAIS, S. T., AND HORVITZ, E. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2005).

[19] VIDSTATSX. Youtube channel, subscriber, & video statistics. `http://vidstatsx.com/`.

[20] WU, B., AND DAVISON, B. D. Identifying link farm spam pages. In *Proceedings of the Special Interest Tracks and Posters of the 14th ACM International Conference on the World Wide Web* (2005).

A Appendix

Here we provide more details regarding the actual exploit and test corpora for the search personalization attack.

A.1 Search Term Variance

As with the various product categories on Amazon, it is reasonable to expect that the effectiveness of search history pollution depends on the value of the search term being polluted. In other words, just as Amazon tightly controls the gift cards it recommends, it might be the case that a website cannot be promoted in Google's search results as easily for a highly competitive search term, such as "laptop", as it can for relatively uncontested search terms. To obtain an estimate of the value of different search terms, we again turned to Google's AdWords Keyword Tool. The tool provides a function that associates a given search term with a level of competition. The competition level is a measure of how expensive it would be for URL to consistently pay enough to be ranked at the top of the list of advertisers for a particular search term. Competition level is expressed as a value from 0 to 1, with 0 having no competition and 1 having fierce competition.

Recall that out of the 2,136 webpages that we attempted to promote using a contextual pollution attack, 729 were successful. It is important to note that some of the promoted results were for the same initial search terms. Therefore, the number of search terms associated with the webpages are 1,740 and 606, respectively. As an example, we attempted to promote both `made-in-china.com` and `DHgate.com` with respect to the original search term "watch". The keywords injected by the pollution attack differ, however, and are "China" and "China wholesale" respectively. For the persistent attacks, we were successful in promoting at least one returned website for 247 out of the 551 search terms.

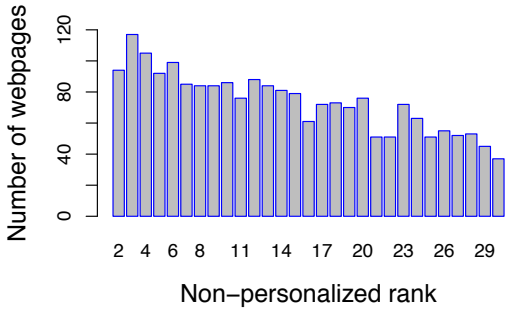


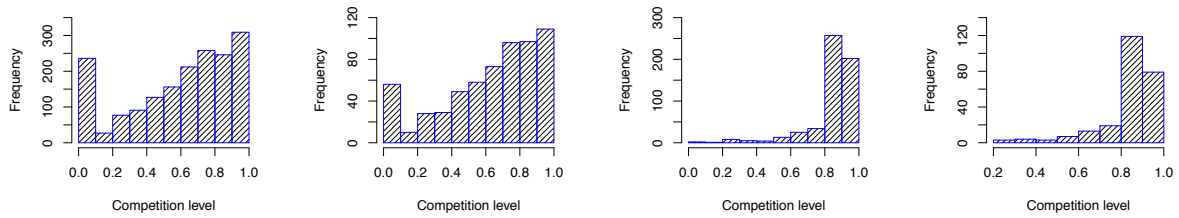
Figure 11: Google's original rank distribution for the 2,136 webpages whose ranking we attempt to improve with contextual search history pollution.

Figure 12 shows the competition level distribution for both types of attacks. Figures 12(a) and 12(b) correspond to the 1,740 search terms associated with our entire contextual test corpus and the 606 search terms for which there was a website we could promote. Likewise, Figures 12(c) and 12(d) plot the competitiveness of the search terms for the 551 tested and the 247 successful persistent pollution attacks. Although the distributions are different between test corpora, in both cases, the distributions suggest there is no obvious correlation between search term competition or value and the likelihood of being able to launch a search history pollution attack.

A.2 Robustness

Because a contextual history pollution attack uses only a few recent search history entries to promote a website, the lifetime of this attack is limited to the period when Google's personalization algorithm considers this contextual information. We empirically determine Google's timeout threshold by injecting sets of contextual keywords into a Google search profile and then pausing Google's history collection. We then search alternatively for two distinct search terms—one that we know is affected by the injected keywords, and another we know is not. We continue to search for these two terms, recording and time stamping all the search returns.

Our analysis of many such tests with different sets of search terms indicates that Google appears to enforce a ten-minute threshold on context-based personalized search, which thereby limits the scope of the contextual pollution attack. Similarly, there are limits on how many different searches can be conducted before the



(a) Entire corpus, contextual (b) Successful attacks, contextual (c) Entire corpus, persistent (d) Successful attacks, persistent

Figure 12: Distribution of search-term competition levels.

injected context is no longer used to personalize subsequent queries. Our initial testing indicates that personalization falls off after the fourth search. Hence, we conclude that the pollution attack can last for at most four subsequent queries or ten minutes, whichever comes first.

Our testing of persistent attacks shows that if a webpage remains promoted after several search terms, it will remain promoted for a long time. To determine how long, we identified a set of 100 webpages and search terms on which we launch a successful persistent pollution attack. We then inject additional randomly selected trending keywords one-by-one and continually check whether the promotion remains. 72% of the websites remain promoted after 60 additional keywords, indicating that, when successful, persistent pollution attacks are likely to remain effective for quite some time.