MASTER IN COMPUTER SCIENCE

Detection of Cybersquatted Domains

Master Thesis

Patrick Frischknecht

from

Laupen BE, Switzerland

Faculty of Science
at the University of Bern

31 July 2021

Prof. Dr. Oscar Nierstrasz

Pascal Gadient

Software Composition Group
Institute for Computer Science

University of Bern, Switzerland

b UNIVERSITÄT BERN

UNIVERSITÉ DE NEUCHÂTEL

UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Abstract

Domain names, or short domains, are memorable identifiers for websites, however their affiliation is not always clear. Cybersquatters register domains that closely resemble existing ones or well known trademarks for their own profit and therefore misuse the trust of a brand. The focus of this thesis is to support security personnel in the accurate detection of cybersquatted domains. Our goal is to identify such domains that have been crafted in bad faith based on the content present on the website, and therefore effectively reduce the number of websites that would otherwise require a manual review. We developed a tool based on logo matching with image hashing that can, given a target domain, report cybersquatted domains in global-scale domain lists that consist of several hundred million entries. For our case study we selected the websites of nine well known luxury and apparel trademarks from the Forbes Top 100 most valuable brands list that we fed to our tool. We performed a manual evaluation on more than 5 000 reported websites to determine whether the automatically assigned label, harmless or malicious, was correct. We realized that cybersquatting is still a relevant issue for selected brands as they try to protect themselves against this threat. Furthermore, we could identify 1 433 domains that host malicious content, including 639 fake web shops. Finally, we realized that image hashing algorithms are preferably not used in such scenarios, because logos on squatted domains are altered in a way that causes large differences in their similarity scores although they remain visually similar. We conclude that logos are indeed a typical feature used in many websites of cybersquatted domains and that our tool can report domains missed by existing tools and services.

Contents

1 Introduction

Cybersquatting is the abusive registration of a domain name confusingly similar or identical to a well known trademark.[1] Cybersquatters, *i.e.*, people who perform cybersquatting prepare websites on squatted domains to trick users into thinking that they interact with an original website, but instead they interact with a counterfeit. Cybersquatters try to profit from the reach and reputation of the trademark owner, *e.g.*, by selling ads, counterfeit products, or by stealing credentials. Another common practice is to sell the squatted domain back to the trademark owner.

Unfortunately, the protection of trademark owners and internet users against cybersquatted domains is not trivial. More domains are registered every day and an independent authority would need to assess each domain in combination with the content to determine whether its setting is legitimate, or if it was registered in bad faith. Even worse, the entire content of a website can change within seconds to spoof potential investigators. Therefore, various national and international laws and institutions have been created to help in protecting trademark owners, *e.g.*, the *World Intellectual Property Organization (WIPO)*, which offers to mediate internet domain name disputes.[2] Despite these efforts, cybersquatting remains a prevalent issue, *i.e.*, the WIPO reports that from January through October 2020 they handled 3 405 cybersquatting cases, which is an increase of 11% compared to the same period in the previous year.[3]

[1] https://www.govinfo.gov/content/pkg/CRPT-106srpt140/html/CRPT-106srpt140.htm
[2] https://www.wipo.int/amc/en/domains/
[3] https://www.wipo.int/pressroom/en/articles/2020/article_0026.html

In this thesis, we present an effective and scalable approach to identify squatted domains that have been created in bad faith. Moreover, we have implemented a tool that automates that process, and which only requires the existing domain name of interest, a set of trademark logos, and a list of registered domains.[4]

We examine the following two research questions with respect to cybersquatting:

- **RQ**$_1$: *Can we efficiently find candidates for cybersquatted domains on a global scale?*

  By applying various squatting techniques to the provided target domain name and matching it against the set of all registered domains, we can create a subset that only contains the relevant candidates. We created a tool to explore whether this process is feasible in practice for global scale domain lists. During the implementation of our tool, we found that, as expected, the detected domains include many irrelevant entries due to the simple matching approach, and thus a verification step is required to reduce the number of false positives.

- **RQ**$_2$: *Are logo images a useful feature to detect cybersquatted websites, and if yes, how successful is such a tool in identifying cybersquatted domains?*

  For each relevant domain, the tool downloaded all website content and matched logo images against those provided by the user. Domains that use similar logos were flagged for a manual review. In order to assess the tool performance, we manually analyzed more than 5 000 websites including those that have been flagged and we verified whether the tool labelled them correctly. We found that our tool is able to detect suspicious pages using logo detection for each luxury and apparel trademark used in our analysis. In retrospect, the used image hashing approach was not a perfect fit for our setup, because the algorithm failed to detect many altered logos or those that were only visible in product images.

In summary, the proposed tool simplifies the process of finding cybersquatted domains and is able to find websites that have not been discovered by existing tools or services.

The remainder of this thesis is structured as follows. We provide the required background in chapter 2, and we discuss the implementation in chapter 3. The design of our case study as well as the results are presented in chapter 4. We summarize the threats to validity in chapter 5 and conclude this thesis in chapter 6.

[4]Such lists are available from various operators, *e.g.*, Zonefiles.io.

2 Background

Cybersquatting has been a major topic in both security and legal research. The focus of this section is to present and discuss the results of previous research that guided our work.

2.1 Definitions

The term (cyber)squatted domain is commonly used to refer to domain names confusingly similar to a trademark or domain owned by someone else. By comparison, the term lookalike domain focuses rather on the visual aspect of a website and not its purpose or domain name similarity, however it is often interchangeably used in the literature. Therefore, without context it is often unclear whether a discussion about lookalike domains addresses legitimate websites or websites built with a malicious intent.

In order to improve clarity on the implications of such terms, we present the following definitions for terms that we use in this work:

- *Target trademark*: The trademark of interest that requires protection.
- *Target domain*: The corresponding domain of a target trademark.
- *Lookalike domain*: A domain name that looks confusingly similar to a target domain without considering any website content.

- *(Cyber)squatted domain*: A lookalike domain that has been registered or used in bad faith in order to profit from the reputation of the target trademark.

2.2 Squatting Techniques

Cybersquatters use a wide range of techniques to create a domain name that can be confused with a target domain. Understanding the different types of squatted domains is key to detecting them. Consequently, we present in the remainder of this subsection four squatting techniques that have been commonly mentioned in the literature.

2.2.1 Typo-squatting

A typo-squatted domain uses a domain name that suffers from one or more potential typos compared to the target domain. This category of domains is constructed with the intent that if a user accidentally mistypes a target domain the squatted domain is reached instead. For example, to target the domain `facebook.com`, a cybersquatter could register `gacebook.com`, which would be reached if a user accidentally types the letter `g` instead of `f`. This is not unlikely since these letters are next to each other on a typical keyboard. In 2013 over 95% of the 500 most popular websites were actively targeted by cybersquatters using typo-squatted domains [3].

2.2.2 Homograph-squatting

Homograph-squatted domains abuse the fact that certain characters or character combinations are barely distinguishable. The intention behind this category of domains is that users will likely not notice any difference when they access them. They are primarily distributed as clickable links through digital communication channels such as email, because the required characters are usually difficult to input manually using another keyboard layout. For such domains, the cybersquatter replaces characters in the domain by other similar looking characters, *e.g.*, “o” looks similar to “0” thus a cybersquatter could register `g00gle.com` to mimic `google.com`. Moreover, the introduction of *Internationalized Domain Names (IDN)* enabled cybersquatters to use non-Latin characters as a replacement for similar looking Latin characters. Such an attack is typically referred to as *IDN homograph attack*. To protect users from IDN homograph attacks browsers employ strict rules about the allowed scripts and individual characters when a domain is translated to Unicode.[1,2] Whenever such rules are violated, browsers only display its punycode[3] in their address bar. However, confusable characters within the same script are typically allowed and attackers can also exploit other weaknesses in these rules to create convincing lookalike domains.

[1]`https://www.chromium.org/developers/design-documents/idn-in-google-chrome`

[2]`https://wiki.mozilla.org/IDN_Display_Algorithm`

[3]A punycode of an existing domain is the Unicode formatted domain name represented with the default ASCII character subset. For example, `google.com` with the Cyrillic letter “e” corresponds to the punycode `xn-googl-3we.com`.

2.2.3 Combo-squatting

In combo-squatting the target domain is combined with additional words or letters like `facebook-login.com` for the target domain `facebook.com`. Attackers use this category of domains to mislead potential victims because the authority validation of a specific website is a non-trivial task. According to Kintis *et al.*, the attackers carefully craft combo-squatting domains for selected target trademarks to maximize their impact [8]. Since some original trademarks use multiple distinct domains for different, related services such attacks are particularly difficult to detect for a typical web user. Kintis *et al.*, for example, complained that *Paypal* uses the domain `paypal-prepaid.com` to advertise its prepaid debitcards, instead of using a subdomain like `prepaid.paypal.com` or a filepath such as `paypal.com/prepaid`. According to them, this trains a user into believing that each domain containing a particular trademark is legitimate. Combosquatters typically use words that are related to the business of the target domain, for example *free* and *cheap* for online shopping pages. Other sources of words for combo-squatting are general internet terms like *login*, *activate*, *online*, or *mail*. Both Kintis *et al.* and Tian *et al.* identified combo-squatting as the most common cybersquatting type in large scale studies [17].

2.2.4 Top-Level Domain-squatting

A very simple squatting technique is the use of a different Top-Level Domain (TLD) that has not yet been registered by the target trademark owner. For example, considering the website `chase.com` of the bank Chase, which is one of the top ten largest banks globally, one could register `chase.global`, which is currently available. Both country code TLDs (ccTLDs) like `.cn` and generic TLDs (gTLDs) like `.com` are regularly used by domain squatters. However, the large and growing number of gTLDs requires more effort from the trademark owners to adequately protect their intellectual property [7]. TLD-squatting shares many similarities with combo-squatting: the domains mislead customers with a pretended and hard to verify authority.

The presented overview about the available squatting techniques in this subsection is not complete, *i.e.*, it presents only the most relevant techniques found in the literature. There exist more techniques, *e.g.*, abbreviation-squatting which leverages short forms of existing domain names [12], or bit-squatting that simulates random bit-errors in memory to form the squatted domain name [14]. Ultimately, the existing techniques can be combined. For example, a perpetrator could add an additional word to the target domain (combo-squatting), then replace a single letter with a lookalike (homograph-squatting) and finally use a different gTLD than the target domain (TLD-squatting).

2.3 Usage

Squatted domains can be employed in a large variety of attacks, *i.e.*, they increase the chance that a user is fooled. Generally speaking, all squatted domains are involved in some kind of *trademark abuse* since the perpetrators use the reputation and reach of a trademark to gain advantages without the explicit consent of the trademark owner. However, researchers have identified five distinct areas within this topic:

2.3.1 Domain Parking

The primary purpose of a parked domain is to serve a page loaded with ads and no real content. The squatted domain name is solely used to attract traffic that eventually generates ad revenue [19].

2.3.2 Domain Selling

A cybersquatter registers domains similar or identical to well known trademarks to later sell them with a profit to the company owning the trademark. At least, these websites declare information on how to purchase the domain name. The lines between domain parking and domain selling are blurred: most squatted domains simultaneously leverage both practices [20].

2.3.3 Affiliate Marketing

Affiliate marketing is a form of advertisement in which an online retailer pays a commission to an intermediary (*i.e.*, the affiliate) based on the sales or traffic that are generated through the referral. In order to identify the correct affiliate a special link with usually one or more additional query parameters is used. Upon visiting this link, an *affiliate cookie* is set in the user's browser that uniquely identifies the affiliate. From the retailer's perspective, affiliate marketing can be abused in several ways, *e.g.*, cybersquatters can use squatted domains to redirect users to target domains that support affiliation programs and in the process attach an affiliate identifier to the request. In the end, the cybersquatter is eligible to receive the commission whereas from the retailer's perspective the cybersquatter did not generate any additional traffic since the user already intended to visit their web shop in the first place. There exist other methods to abuse affiliate marketing systems, however domain squatting seems to be one of the most prevalent. In a recent study, web surfers received 84% of all *affiliate cookies* through typo-squatted domains [6].

2.3.4 Phishing

Phishing websites are used to steal user credentials by mimicking the look and feel of a target trademark's trusted website. Squatted domains can increase the likelihood that a user enters some credentials, however the number of active phishing domains within all squatted domains is rather small. That is, in a large-scale analysis of over 600 000 squatted domains only 1 741 (0.2%) could be attributed to phishing [17].

2.3.5 E-commerce Fraud

The two typical forms of e-commerce fraud are counterfeit web shops that sell illegal replicas and fake web shops that sell goods and services without any real intention to deliver. A squatted domain is no necessity to commit e-commerce fraud, but they are widely used for this purpose. Such exploitative web shops cause substantial damage to both their unaware customers buying inferior goods and the affected trademarks for not generating any revenue. Fake web shops using squatted domains have been observed in a large scale study without being the main focus of the work [8].

2.4 Related Work

Domain-squatting has been a major topic in security research for many years now. Companies and the open-source community have developed services and tools to detect and fight squatted domains. In this section, we present recent research and tools related to cybersquatting as well as logo matching, a technique that we employed to reduce the required manual effort when using such a tool.

2.4.1 Cybersquatting

Kintis *et al.* performed a large-scale study using over 468 billion DNS records from both active and passive DNS datasets to investigate how prevalent combo-squatting is and how it is used [8]. They found over 2.7 million combo-squatted domains targeting one of the 268 selected domain names from the top 500 in the Alexa Ranking in the United States.[4] Similarly, Agten *et al.* monitored a large list of pre-generated typo-squatted domain names that targeted any of the Alexa top 500 domains over a period of seven months [3]. They discovered that over 75% of all short, typo-squatted domain names for these popular domains had already been registered. Tian *et al.* focused specifically on squatted phishing domains [17]. Using an optical character recognition technology and other textual and content-based features they could identify 1 224 phishing web pages from over 224 million DNS records, of which they could confirm 1 175 as real phishing pages during their manual analysis. Identical to these studies, our tool works on a global scale and considers the most relevant kinds of domain-squatting including combo and typo-squatting.

2.4.2 Tools and Services

There exist several well-known and publicly accessible tools that can report squatted domains. *dnstwist* is a widely recognized open-source tool used to detect squatted domains.[5] It uses various techniques to create lookalike domains and then queries each of them to check whether they are in use. To detect potential phishing pages it computes a fuzzy hash on a website's HTML content and compares it to the hash of the target site. Another well-known tool is *SquatPhish* that is built on top of *dnstwist* and adds limited support for combo-squatting and improved the homograph detection capabilities [17]. Lastly, *openSquat* supports bit, homograph and typo-squatting, and uses the *Levensthein* word distance measure combined with additional logic to accurately detect them.[6] Moreover, there exist services that can report squatted domains. For example, Facebook offers a certificate transparency monitoring service to warn users about newly registered domains similar to their domains to help them combat phishing.[7] Advancing the reported tools and services, our tool has abilities that are currently lacking in these tools such as the support for combo-squatting detection and visual similarity features.

[4] `https://www.alexa.com/`
[5] `https://github.com/elceef/dnstwist`
[6] `https://github.com/atenreiro/opensquat`
[7] `https://developers.facebook.com/tools/ct/subscriptions/`

2.4.3 Image Matching

In prior work, image matching has been widely used to detect phishing and other types of online fraud. For example, *Verilogo* uses the SIFT algorithm [11] to find logos in website screenshots to detect potential phishing sites [18]. Afroz *et al.* followed that idea and created a tool that lets its users label protected sites prior to the analysis such that the algorithm can compare the content, especially images, of potential phishing web pages against the content of the previously stored trusted site [2]. *LogoSENSE* employed the Histogram of Oriented Gradients (HOG) feature descriptor to detect brand logos on web page screenshots in a scale invariant fashion similar to the SIFT algorithm [5]. Finally, Zhao *et al.* reviewed how computer vision methods were used in network security. In their study they present a comprehensive overview of logo detection-based approaches [21]. Another form of image matching is image hashing. This approach assigns every image a sufficiently unique bit string that remains close if two images are visually similar. This technique is very efficient since the source images are not anymore required after the unique bit string has been determined for every image that should be compared. Additionally, it requires no training data specific to each target trademark unlike approaches based on neural networks. Consequently, it has been used in various domain-squatting studies. For instance, Tian *et al.* applied image hashing to the screenshots of phishing pages [17]. Based on the difference between the hash of the original site and the phishing site they showed that domain-squatting employed in phishing attacks sometimes modify the original layout drastically to escape detection. Similarly, Kintis *et al.* used perceptual image hashing to detect phishing pages among all squatted domains that contain any kind of login forms [8]. Similar to existing work, we use image hashing to match logos on suspicious lookalikes domains since we expect them to be a prevalent feature on all kinds of malicious websites that try to mimic the look and feel of the original website.

2.5 Detection Strategies

Various techniques have been developed to detect lookalike domain names corresponding to the presented squatting techniques. Therefore, we discuss in the remainder of this subsection three common detection strategies and the relevant tools.

2.5.1 Generative Techniques

Generative detection techniques detect lookalikes domains by comparing a domain to a set of pre-generated candidates that were created using rules on how to produce potential squatted domain names given a target domain. For example, the generation of homograph-squatted domains requires at least one character to be replaced by a different but confusable character. More generally, by analyzing existing squatted domains one can formulate generative models that, given a target domain, are able to create a finite list of potentially squatted domain names. With such lists, deciding whether a domain is a lookalike of a target domain is a rather simple task: if a lookalike is present in one of the generated lists it is considered a squatted domain name. Researchers have already extracted numerous generative models. In particular, Wang *et al.* identified five different models, *e.g.*, like character-replacement and character-insertion, which

Target trademark	Partially matching domain name
Nike	median-kliniken.de
Amazon	amazonas.eu
Apple	applejuicedesign.com
Ikea	mikealodge.com
EA (video game company)	beyondmeat.com, buildabear.com

Table 2.1: Trademarks unfit for partial string matching techniques and samples of problematic matches.

they support in their typo-neighborhood generator tool to create a list of typo-squatted domains [19]. Furthermore, Tian *et al.* found and implemented generative models for homograph-, typo-, bits-, and TLD-squatting in *SquatPhish*[8] an open-source tool to identify squatted domains [17]. Their tool is based on *dnstwist*, a popular and actively maintained open-source Python library and command line tool for domain name permutation.[9] While these approaches indeed support a large variety of squatting techniques, a generative model is insufficient for combo-squatting, because there usually exists no reasonable, finite list of combo-squatted domain names for a given target domain, due to the large number of words that could be combined with the target domain. As a workaround *SquatPhish*, for example, considers all domains that contain the target domain and a hyphen as combo-squatted domains. Unfortunately, this approach still misses combo-squatted domains that use different or no characters at all for the word separation.

2.5.2 Distance-based Techniques

String edit distances, *e.g.*, *Damerau–Levenshtein* can be used to identify squatted domains. They compute the number of edit actions required to transform an existing word into the desired word, *i.e.*, the smaller the edit distance the few changes are required for the transformation of the word. Domains are usually considered lookalikes when they have an edit distance below two. Edit distances can detect typo-squatted domains as shown by Moore *et al.* who developed a new edit distance metric called *fat-finger distance* [13]. This metric takes into account that adjacent letters are more likely to be typed accidentally than other letters further away. As a result, the string edit distance is a part of *Google Chrome*'s heuristics to warn its users about potential lookalike domains.[10] However, a major downside of string edit distances is their incompatibility with combo-squatting, *i.e.*, an additional word drastically increases the distance and renders their use unsuitable.

2.5.3 String Matching Techniques

Another technique used to detect combo-squatted domains is partial string matching. According to this technique, a domain name is considered a lookalike if it contains (sub-)strings of the target trademark. Whereas this process usually works well for long and distinct trademarks it may reports lookalikes that most

[8]https://github.com/SquatPhish/1-Squatting-Domain-Identification
[9]https://github.com/elceef/dnstwist
[10]https://chromium.googlesource.com/chromium/src/+/refs/heads/main/docs/security/lookalikes/lookalike-domains.md

users would consider unrelated to the target trademark for shorter trademarks, or trademarks composed of words commonly present in dictionaries, *e.g.*, “is”, “apple”, *etc.* To illustrate the problem, Table 2.1 presents some examples of problematic trademarks and the erroneously reported lookalikes. In order to mitigate this problem, researchers have tried to exclude trademarks that are exceptionally short or composed of common English words [8].

2.6 Verification

After a lookalike domain has been identified the trademark owner may want to know whether it is registered or used in bad faith to avoid initiating futile measures that are time demanding and expensive against a legitimate domain. Whereas the affiliation of certain reported domain names is clear by just reading them, *e.g.*, `applejuice.com`, it can be very difficult for others without looking at their content, *e.g.*, `applebank.com` that could possibly relate to *Apple Pay*. In the remainder of this subsection we briefly cover conventional measures to identify fraudulent websites.

2.6.1 Text Similarity

It is very common for squatted domains to reuse large portions of the target trademark’s website, *e.g.*, the text and the HTML markup to delude potential victims. Several approaches use this fact to detect fraudulent websites. For example, the free tool *dnstwist*[11] generates a fuzzy hash value[12] of the website’s HTML code and compares it with the one from the original domain. A high similarity of both values indicates that the content of the lookalike domain is very similar and most likely a phishing page. Instead of text hashing, Tian *et al.* extracted text features from both the HTML code and a screenshot of the website using *Optical Character Recognition (OCR)* technology [17]. Based on these features they trained a random forest classifier to label about 600 000 squatted domains for 702 different brands with which they could confirm about 1 175 (67%) of the reported phishing pages in their manual evaluation.

2.6.2 Logo Similarity

Logos are key to a brand’s online identity, *i.e.*, most websites use the corporate logo to reveal their affiliation. In consequence, impersonating websites usually abuse the logo of the targeted trademark. Although the detection of logos was successful in identifying phishing pages [1, 2, 18], to the best of our knowledge, it has not been used in research to verify squatted domains. In practice, the adapted algorithms from the field of computer vision faced three major challenges: i) the accurate detection of a logo among other content in a screenshot or the embedded images of a website, ii) the identification of the targeted trademark and the specific logo that has been mimicked, and finally, iii) the similarity rating of both logos and the final decision about their authenticity. Several approaches have been proposed. For example, *Verilogo* scans a website [18] in stripes of a predefined pixel height. The content of each stripe is

[11] `https://github.com/elceef/dnstwist`
[12] `https://github.com/DinoTools/python-ssdeep`

matched against a predefined list of logos using the SIFT algorithm [11] which leverages features resistant to image distortions. Furthermore, there exist commercial brand monitoring and protection services based on computer vision, *e.g.*, provided by *Visua Labs*.[13]

2.6.3 Behavior and Style Similarity

Researchers found that lookalike domains often share resources beyond logos with the target domain that let them look and feel genuine. Such studies commonly leveraged a combination of manual and automated procedures to identify lookalike domains with similar behavior and style but malicious intentions, *e.g.*, performing affiliate abuse or phishing [8]. The researchers proposed a perceptual hashing function to cluster visually similar login websites to reduce the required manual effort for the subsequent analysis tasks. In general, existing approaches to detect lookalike domains are very similar to those used for phishing detection, *i.e.*, the major difference is their additional logic to distinguish specialized scenarios like affiliate abuse. Unfortunately, comprehensive tool support to investigate lookalike websites at large is still largely missing.

2.7 Domain Lists

Domain lists that resemble a snapshot of the current domain name registrations along with additional metadata, *e.g.*, the used name servers are essential to efficiently find squatted domains. They are a crucial component for all non-generative detection approaches as they represent the dataset in which squatted domains can be identified using various matching techniques. Such domain lists can be obtained from free providers, *e.g.*, the *Active DNS project*[14] which holds records for the most common TLDs `.com`, `.name`, `.net`, `.org`, and `.biz` [9]. Another free source of registered domain names are the certificate transparency logs.[15] These logs contain the history of used TLS certificates and are essential for HTTPS-protected websites, because web browsers recently started to demand for certificates that have been published in such logs in order to be accepted. As a result, one can monitor these logs to collect all domain names used in newly issued certificates. In addition, there exist a plethora of commercial providers that sell more comprehensive domain lists that are regularly updated, *e.g.*, *ZoneFiles.io* and *WhoIsDataCenter.com*.

[13]`https://visua.com/the-benefits-of-visual-ai-in-brand-protection/`
[14]`https://www.activednsproject.org/about.html`
[15]`https://datatracker.ietf.org/doc/html/rfc6962`

3 Implementation

To verify if a particular lookalike domain is used in a malicious habit one must manually search for common threats associated with domain-squatting which is a time demanding process. Therefore, we implemented a tool that can accurately detect squatted domains autonomously at large using logo matching. In this section, we begin with the discussion of our objectives, before we explain the tool's detection process and its architecture in more detail.

3.1 Objectives

Combo-squatting. To improve upon existing solutions the implemented tool should detect *combo-squatted* domains. Those are often ignored since they introduce numerous false positives. However, they are the most common type of domain-squatting.

False positives. Websites that could cause false positives should be removed early in the process before logo matching is applied. For example, sites that perform *affiliate abuse* or *parked domains* usually do not use any trademark logo in their content and can be safely excluded from further analyses. To detect such sites, indicators like unusual query parameters, or the lack of content except ads should be used.

Performance. Performance is another major concern. To support large domain lists and to maintain a low cost for the analysis of each suspicious squatted domain the tool should work efficiently, and it should be easy to scale up.

Ease of use. In general, the tool should require only a few user inputs and be easy to use. The tool will require a list of registered domain names, which can easily be acquired from the internet. The logo matching is implemented using an image hashing algorithm that only requires a few sample logo images per target trademark.

Report. At the end of the execution the user should receive a short and simple report that should be manually verified to identify the relevant threats for the target trademark. The ability to manually verify the results is important since we do not expect that our tool will convincingly work out of the box for every scenario due to the different requirements of business categories. We will not implement any technique to exclude harmless lookalikes using logos, *e.g.*, other pages from the same company since the decision whether a use is legitimate is very challenging.

3.2 Detection Process

Figure 3.1 provides a high-level overview about the workflow of our tool used for accurately detecting the squatted domains. In the figure, inputs required by the user for each execution are presented with a yellow background, and the labels that we assigned are presented with a blue background. We closely follow the presented flow chart in this subsection. Before starting the process, the user has to provide three major inputs:

- A list of registered domains (**Domain List**)
- The domain name of the target trademark, *i.e.*, the target domain (**Target Domain**)
- A set of representative logos for the target trademark (**Logo Images**)

The list of registered domains is split into smaller chunks to leverage concurrent executions to reduce the overall analysis time.

3.2.1 Detection of Lookalike Domains

To achieve a high performance, the number of domains that require a detailed analysis must be reduced as much as possible without sacrificing the accuracy of the results. We achieved that by excluding "sufficiently different" domain names early in the process that we consider irrelevant. To determine "sufficiently similar" domain names we leveraged existing libraries, *e.g.*, the *dnstwist* library that supports a variety of different squatting techniques including typo and homograph-squatting. Given a target domain it can automatically generate large lists of potentially squatted domain names. We consider a domain to be a lookalike if a domain name generated by dnstwist without the TLD exactly matches any of the subdomains present in the registered domains list. We never match any TLDs to ensure that Top-Level Domain-squatting is correctly detected, which is unsupported by dnstwist. Moreover, dnstwist lacks support for combo-squatting that we had to implement ourselves using partial string matching techniques. At first, we dissect the provided

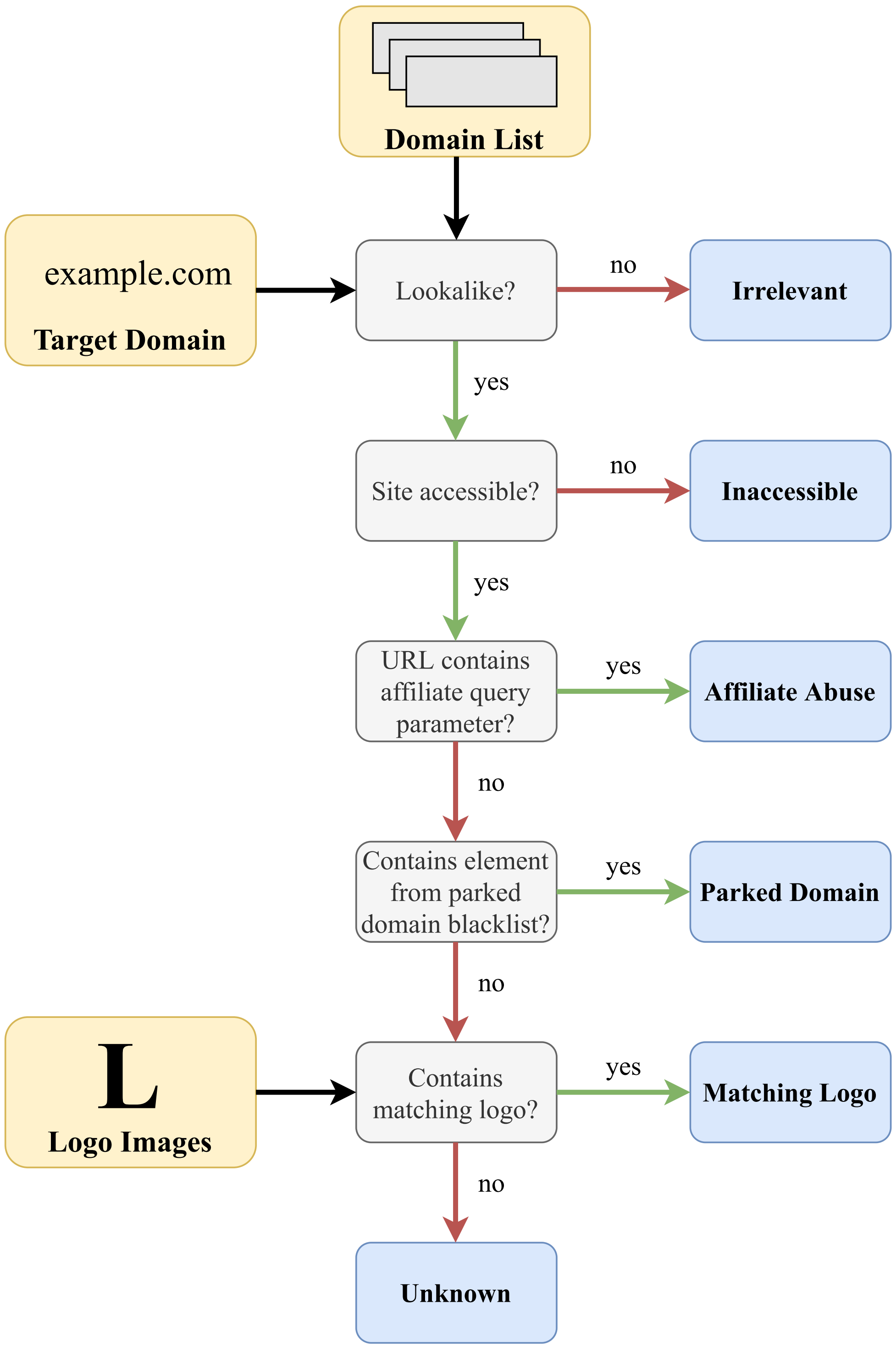


Figure 3.1: The workflow used for the detection of squatted domains followed by our tool

target domain into the TLD component and a list of subdomains, *e.g.*, `en.wikipedia.org` consists of the two subdomains `en` and `wikipedia` of which the latter is also a second-level domain, as well as the TLD `org`. Using the collected second-level domain, we would then match the target domain against all other subdomains in the registered domain list. We consider a domain to be a lookalike if the second-level domain of the target domain is included in any of the subdomains in the registered domains list. If we continue our example and assume that our target domain would be `wiki.de` the tool would match the second-level domain `wiki` against both `en` and `wikipedia`. Since `wiki` is part of `wikipedia` the tool would consider `en.wikipedia.org` a lookalike of `wiki.de`. By applying this algorithm on all registered domains in our data source we receive a list of lookalike domains that require further verification and may contain squatted domains. We consider the domains that are not in this list *Irrelevant*.

3.2.2 Exclusion of Inaccessible Websites

Next, we remove inaccessible websites from our list of interesting lookalikes. For that purpose, our tool uses a remote controlled headless browser that tries to visit each collected lookalike domain and if successful, it collects a screenshot after a short delay to let the rendering process finish, and after that the entire text content in the HTML code. Modern websites heavily use JavaScript code to dynamically add content and thus using a full-fledged browser is a necessity, because otherwise we would risk to obtain incomplete data since the payload contained in the initial HTTP response is insufficient for rendering such views. At the same time, this process evades some mechanisms used by websites to block undesired clients due to the authentic browser information that is attached to the requests which mimic the browser of a regular user. However, there exist still methods to detect the use of an automated browser, *e.g.*, with a JavaScript API. The automated browser follows all redirects before extracting any data. We assigned the *Inaccessible* label to registered domains whose websites were inaccessible, before we removed them from our list of interesting lookalike domains.

3.2.3 Detection of Affiliate Abuse

From the remaining lookalike domains, we first tried to identify domains related to affiliate abuse, because many of them can be identified based on the distinct query keys and values used in such URLs. Therefore, we compiled a list of such unique keys and values by manually analyzing a few dozen different affiliate domains. We found that the query key `tag=` and the three query values `=aff`, `=affiliate`, and `=affilinet` are commonly used in affiliate URLs. We label a domain *Affiliate Abuse* and remove it from our list of interesting lookalike domains if any of the identified keys or values is present in the final URL, *i.e.*, after we followed potential redirects.

3.2.4 Detection of Domain Parking

A second common threat is domain parking. Most parked domains seem to be hosted by a small set of domain parking services like *Sedo*[1] or *GoDaddy*.[2] Figure 3.2 shows some typical examples of parked domains. While looking at these parked domains one can see a common element which contains links to websites with a similar name. To create these links, most of these parked domains use the same JavaScript code provided by Google.[3] Thus we can identify such parked domains by searching for the existence of a HTML script tag for this particular JavaScript file. However, not all parking services use it. To be more generic, our tool uses a blacklist of CSS selectors that describe HTML elements that are typically present on the parking sites from parking services. We label a domain *Parked Domain* and remove it from our list of interesting lookalike domains if it contains any element from our blacklist.

3.2.5 Detection of Logo

After the previous steps, the set of lookalike domains that require further investigations still contains a large number of harmless or unrelated websites. Therefore, we used logo matching as a measure to better separate squatted domains from harmless lookalikes. We label a domain *Matching Logo* if any of the images present on the web page matches one of the provided target trademark's logos. Since the logo matching process is rather complex, we discuss it in more detail in the following subsections.

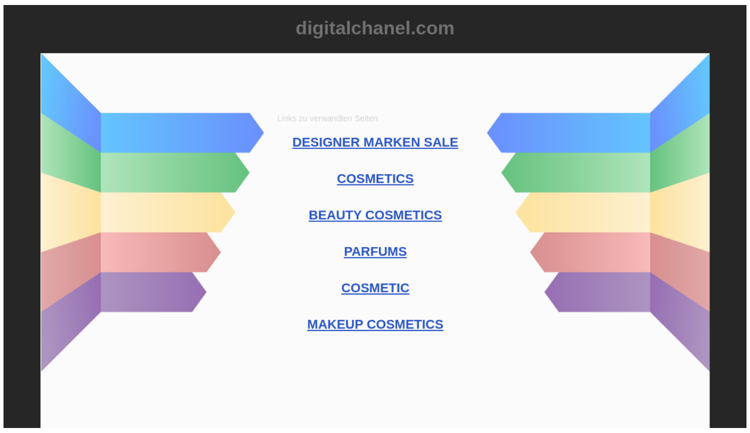
3.2.5.1 Logo Extraction

To perform any image matching the tool must first identify and extract the potential logos from a lookalike domain. Accordingly, our automated browser extracts all relevant images from each visited site. Images can be embedded into a page in various ways using different styles. Our tool considers the following elements: i) `image` tags, ii) `svg` tags, iii) elements with a `background-image` CSS attribute, and finally, iv) a website's favicon that is usually defined in a `link[rel="shortcut icon"]` or `link[rel="icon"]` element in the HTML `head`. If none is found, the tool will try to fetch the icon from the default location `/favicon.ico`.

If one of those elements is found the tool downloads the image from the relevant URL that is usually defined by the `src` property. If the element is of type `svg` the tool transforms the svg source text to a `.png` image file using the `CairoSVG`[4] library. Since this conversion can fail we also take a screenshot of the SVG element on the rendered page in the automated browser.

Images smaller than sixteen square pixels are excluded, but every other image will be used for the logo matching. Therefore, we had to ensure that all pre-processing steps and the matching itself correctly support the various image formats, *e.g.*, `jpeg`, and `png`, that can be present on a website.

[1] `https://sedo.com/us/`
[2] `https://godaddy.com/`
[3] `https://www.google.com/adsense/domains/caf.js`
[4] `https://github.com/Kozea/CairoSVG`


digitalchanel.com

Links zu verwandten Seiten

DESIGNER MARKEN SALE

COSMETICS

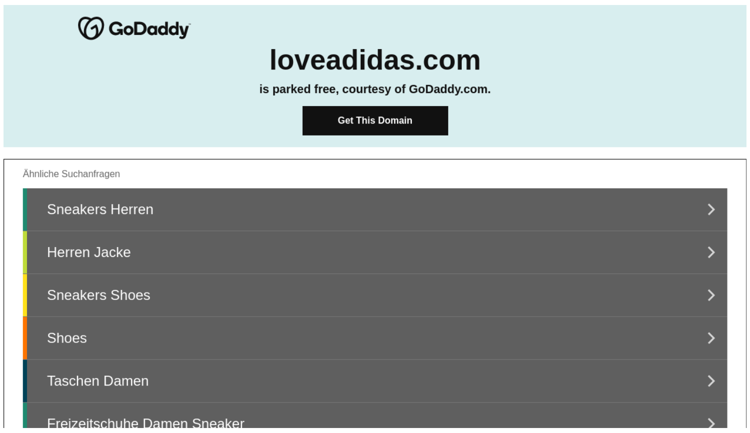
BEAUTY COSMETICS

PARFUMS

COSMETIC

MAKEUP COSMETICS

(a) Example of a parked lookalike domain `digitalchanel.com` abusing the *Chanel* trademark



loveadidas.com

is parked free, courtesy of GoDaddy.com.

Get This Domain

Ähnliche Suchanfragen

Sneakers Herren

Herren Jacke

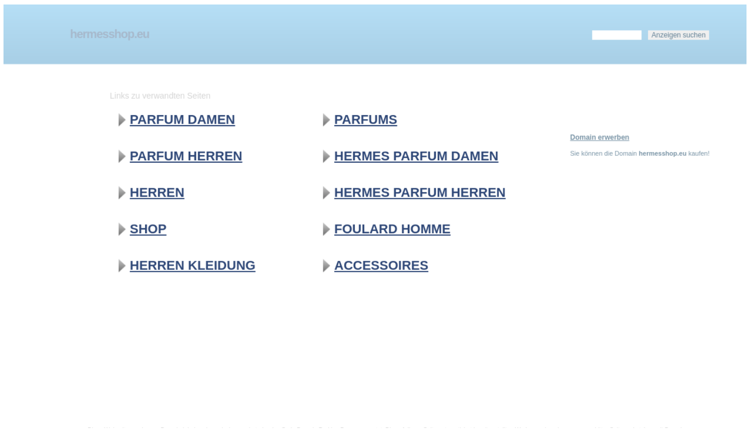
Sneakers Shoes

Shoes

Taschen Damen

Freizeitschuhe Damen Sneaker

(b) Example of a parked lookalike domain `loveadidas.com` abusing the *Adidas* trademark


hermesshop.eu

Anzeigen suchen

Links zu verwandten Seiten

PARFUM DAMEN

PARFUM HERREN

HERREN

SHOP

HERREN KLEIDUNG

PARFUMS

HERMES PARFUM DAMEN

HERMES PARFUM HERREN

FOULARD HOMME

ACCESSOIRES

Domain erwerben

Sie können die Domain **hermesshop.eu** kaufen!

(c) Example of a parked lookalike domain `hermesshop.eu` abusing the *Hermès* trademark

Figure 3.2: Website screenshots of parked domains

A major problem for the subsequent image hashing was transparency, *i.e.*, the alpha channel of an ARGB image. To mitigate this problem, we replace all pixels that have transparency with pixels of a single predefined color that ensures a high contrast compared to the remaining pixels, before we remove the entire alpha channel.

3.2.5.2 Image Hashing

To match the extracted images against the user provided logos of the target trademark a *perceptual image hashing algorithm* is used. Image hashing algorithms take an image as an input and return a short bit string, usually called hash. The algorithm constructs the hash so that two similar images have similar hashes.

Our tool uses the perceptual image hashing algorithm `phash` provided by the `ImageHash`[5] Python library. Given an input image it generates a 64 bit hash when used with the default settings. As a similarity metric the *Hamming Distance* is used, which simply counts the number of different bits with respect to the position within the hashes.

The `phash` algorithm works as follows:

1. Reduction of size and color: the image size is reduced to 32x32 pixels and the image is converted to grayscale.
2. Application of Discrete Cosine Transformation (DCT): a two dimensional DCT is applied resulting in 32x32 coefficients.
3. Removal of high frequencies: 32x32 coefficients are reduced to 8x8 by taking the coefficients for the lowest 64 frequencies.
4. Computation of the median: the algorithm then computes the median coefficient from the previously selected low frequency coefficients.
5. Binarization: each coefficient above the median is mapped to 1, the others to 0. The result is a coefficient matrix of 8x8 bits.
6. Hash creation: the 8x8 bits are reordered to form a 64 bit hash value. The exact order of these bits in the hash value is irrelevant as long as we use the same hash transformation strategy for each image.

`phash` is robust to small changes and compression. We therefore expect that the algorithm should recognize copies from an original logo stored in other image file formats even when they use different compression techniques. Additionally, we apply a few normalization steps to each input image to ensure that simple and common logo image manipulations will not result in a false positive. We elaborate on these normalization steps in the following subsection. To accelerate the image comparison, we pre-compute the image hash for each trademark logo that the user provides. Every image that is found on a lookalike domain will be normalized before its `phash` is computed by the tool. If the Hamming distance between

[5]`https://pypi.org/project/ImageHash/`

the resulting hash and any pre-computed trademark logo hash is below a predefined threshold the tool will label the domain as *Matching Image* and store the match.

3.2.5.3 Normalization

We apply a few normalization steps to each image before the matching process to increase the degree of image manipulations our logo matching approach supports. The normalization ensures that backgrounds and color inversions do not interfere with the image matching. We planned, initially, to apply the image hashing directly without any prior normalization. However, during a test run for the *Netflix* and *Puma* trademarks the algorithm only matched few images. Upon closer inspection, we identified a few common image manipulation techniques that prevented a correct matching. The most common problem was the size and aspect ratio of the image. The squatters converted the original logos, which can often be found as vector graphics in the `.svg` format to `.png` and `.jpeg` pictures with an arbitrary size and aspect ratio, and they filled the background with whatever matched the color of their page, *i.e.*, usually solid black or white. The size and image format change caused no issue, because the algorithm converts and downsizes the image in the process, but the changed aspect ratio resulted in a substantially different hash value since the image after the transformation to 32x32 pixels is very different. We further found that the used color palette of the logos was frequently changed, but this change usually caused no issues since the hash is computed on a grayscale image. However, there is one exception: color inversion. In a typical case the logo is black and on a white background. If the squatted site now uses a dark design, *i.e.*, black background, the squatters invert the logo colors from black to white, and from white to black. This change alters the color distribution and thus the resulting hash value.

To fix these two major issues, we apply the normalization steps shown in Figure 3.3 before we compute the `phash`. The blue border around the images in the figure was added to highlight the image boundaries. A `0x` prefix was added to the 64 bit image hashes, because they are written using the hexadecimal notation. The reported hashes illustrate how image manipulations affect their similarity. In the following, we explain the three normalization steps in more detail:

1. Binarization: We convert the image to black and white only. For this process, we calculate an average grayscale color, *i.e.*, the threshold using the method from Otsu [15]. All pixels above the threshold will be white, the others black.
2. Cropping: We apply an image convolution using a discrete Laplacian kernel. Afterwards, we crop the image to the smallest rectangle that still includes all edges, *i.e.*, all pixels where the result of the convolution is not zero.
3. Inversion: Since the image is binarized we can simply invert the colors by replacing all black with white pixels and vice-versa. Finally, we can compute the `phash` on both the original and the inverted image.

The binarization and the cropping work best on arbitrary complex logos on a background with one color or a high contrast, which is usually the case.

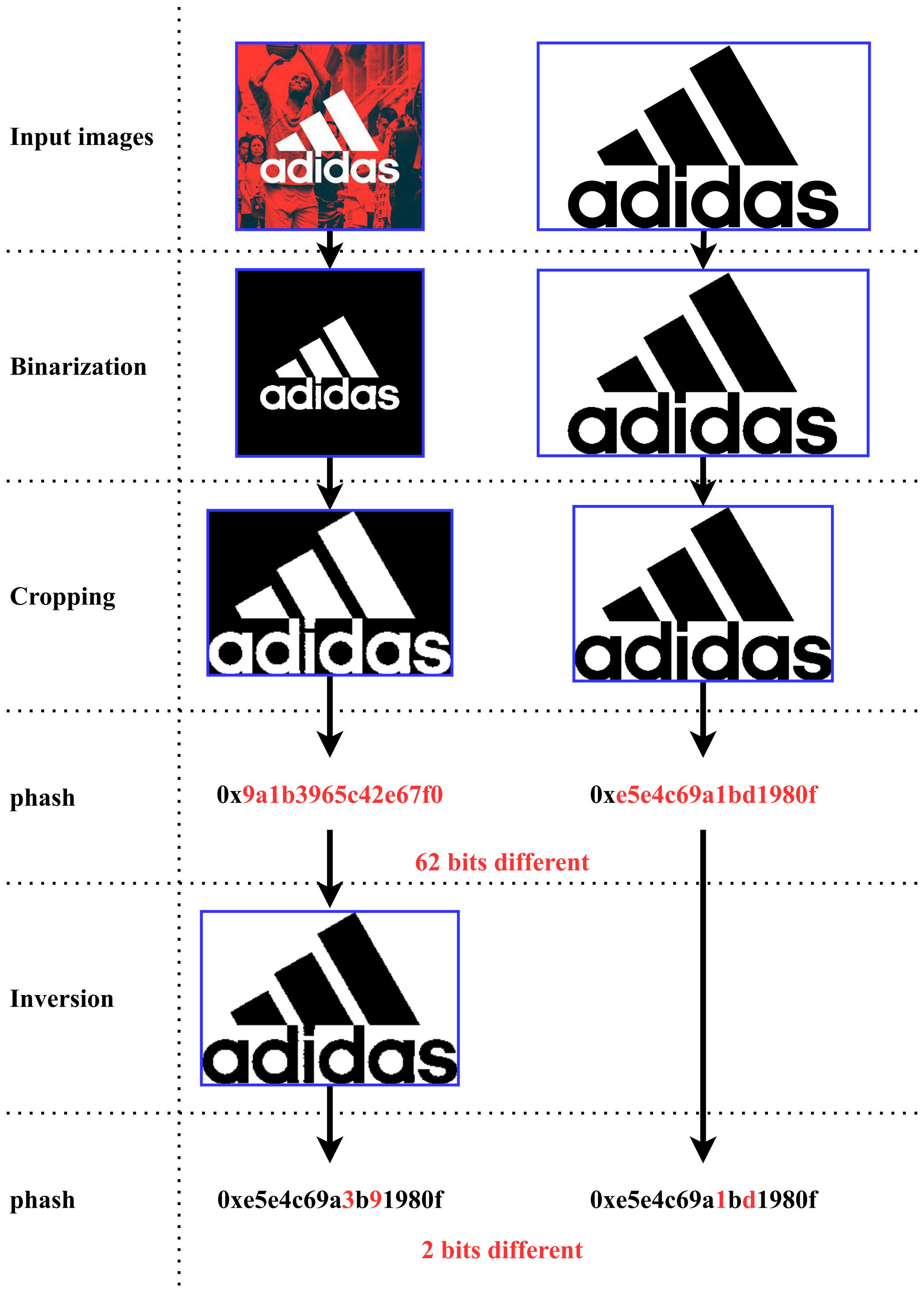


Figure 3.3: Image normalization and matching process

3.2.5.4 Threshold

If the distance between the image hash of two images is below a predefined threshold we consider them a match. This threshold is, consequently, decisive for the quality of the logo matching. The distance between two hashes is reported in bits that are different, the so called *Hamming distance*.[6] In Figure 3.3 we reported the hashes as hexadecimal strings, however they must be converted into their binary representation to compute the Hamming distance.

To determine a reasonable threshold, we used two different datasets for our use case of matching trademark logos against arbitrary web images that could be similar logos. The first dataset consists of 1 000 random web logos from the Large Logo Dataset (LLD) [16], and the second dataset consists of 1 000 random web images from the WebVision 2.0 project that crawled images from Flickr and Google's image search [10]. After we collected these images, we hashed and compared all of them against each other, which resulted in about 2 000 000 comparisons that we performed.

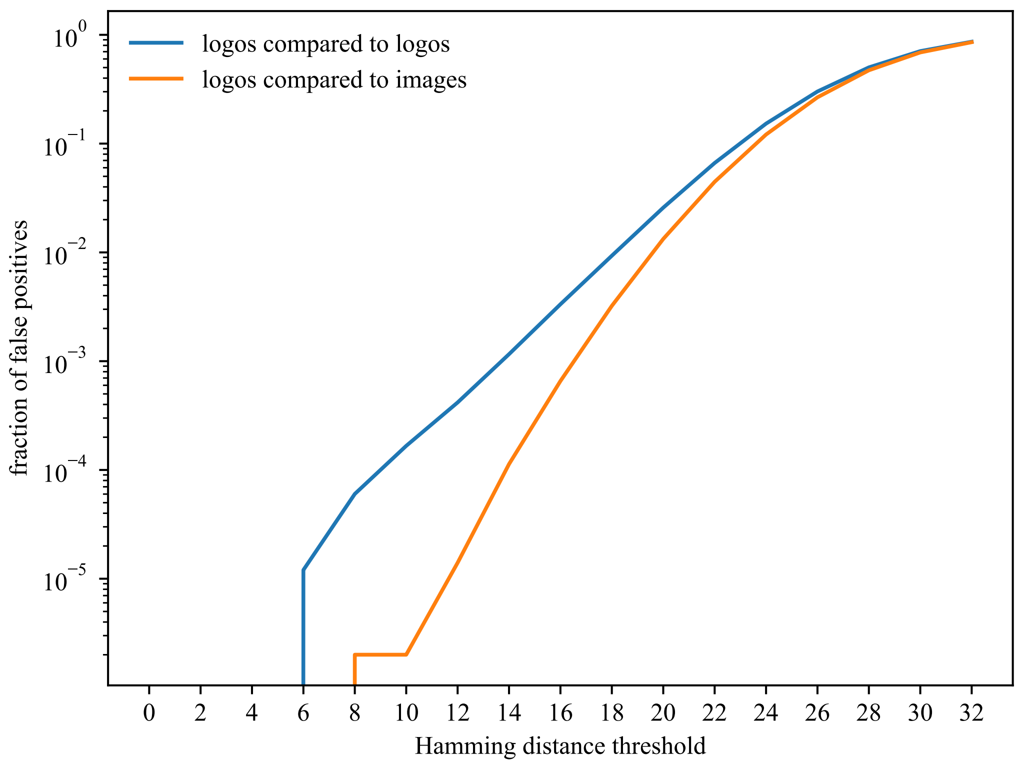


Figure 3.4: Evaluation of different Hamming distances for the similarity threshold

[6] `https://en.wikipedia.org/wiki/Hamming_distance`

A reasonable threshold value should optimize the following two properties: i) the number of matches between different logos should be minimal, and ii) the number of matches between logos and random images should be minimal. The number of matches between different random images is irrelevant since the algorithm always compares a provided logo to an arbitrary website image. The number of matches in either one of the two important categories are depending on the threshold and can be seen in Figure 3.4. The x-axis represents the threshold in bits, and the y-axis represents the share of the false positives, *i.e.*, the fraction of the comparisons that resulted in a match between two images. The orange line represents matches between a logo and a random image, whereas the blue line represents matches between two different logos.

As we can see, there are no matches with a Hamming bit distance that is lower than six and the number of matches remains well below 1% even for threshold values above ten. In other words, image hashing can distinguish logos either from each other or from random images even when the images would undergo minor changes resulting in slightly different image hashes. Ultimately, we chose a threshold value of twelve, because it is the upper bound where the number of false positives for logos compared to other logos remains below 0.1%, what we consider acceptable.

3.2.6 Remaining Lookalike Domains

Even after applying the logo matching some domain names might remain unclassified. We label them *Unknown*. These should mostly consist of domain names that happen to be similar to the target domain, but without posing any threat. Obviously, this category will also contain domains that have not been classified correctly by the previous steps.

In conclusion, every potentially squatted website receives one of the labels *Inaccessible*, *Affiliate Abuse*, *Parked Domain*, *Matching Logo*, or *Unknown*. The tool offers methods to export reports with specific sets of domains, *e.g.*, all domains with the *Matching Logo* label, which then can be manually analyzed by the user. The report is available in both the HTML and the CSV file format and contains, among other things, the domain, the resolved URL, and in case of the HTML version, a screenshot and the matched logo, if applicable.

3.3 Tool Architecture & Deployment

The tool is written in the *Python* programming language. The scripts are executed using *Redis Queue (RQ)*, *i.e.*, a library to queue and process jobs in an arbitrary number of background workers.[7] *systemd* is used to manage the large number of background workers. Each worker follows the procedure described in Figure 3.1. First, we split the initial domain list into small files, then for each of these files we generate a job to let a worker collect the relevant lookalikes and store them as *JSON* objects in a *MongoDB*.[8] Then we partition the lookalike domains in batches of 100 domains. When a worker starts to process such a

[7]`https://python-rq.org/`
[8]`https://www.mongodb.com`

batch, it starts a new *Chrome* browser instance and visits each domain in the batch. Metadata such as the labels, *e.g.*, *Parked Domain* and *Matching Image* are again written as *JSON* objects to the *MongoDB*, whereas screenshots and images are stored in the file system on the hosting server. Since we currently use local storage the tool cannot be distributed across multiple servers without using a centralized storage system, *e.g.*, by mounting a network drive. For our case study, we did not require to distribute the workers to multiple servers; instead we executed them on a single server with 64 cores at which up to 60 worker instances were concurrently running. Since each spawned *Chrome* browser uses multiple threads the entire analysis pipeline leveraged many more CPU cores beyond those that were required for the workers. However, during our analyses we observed that most of the time the majority of workers were idling and waiting for responses from the queried lookalike domains.

4 Case Study

We evaluate our tool with the trademarks of the top nine domains related to luxury and apparels, *i.e.*, fashion accessories to find their cybersquatted domain. For each reported domain, we manually collected more detailed information to discuss the quality of the results. In this chapter, we present the used data sets and the manual review process before we explain the encountered difficulties in identifying cybersquatted domains. Then we report the findings and assess the quality of our tool. We conclude this section with a discussion.

4.1 Methodology

We let the tool determine the lookalike domains of selected target trademarks, and then we manually verified the reported domains. The reported domains include all potential lookalike domains that contain a target trademark's brand name or traces of it in the domain name and the source code of the website. During the manual evaluation, we assign each domain a threat level that indicates the severity, a threat label to indicate the type of threat its website presents, and finally, we denote whether the website contains a logo that belongs to the target trademark. During the manual analysis we focused on domains that belong to the *Unknown* or *Matching Images* category, since these categories will cover the most relevant websites. We only verified websites that contain the original trademark in their text to reduce the number of websites that require a manual analysis, while we strove to include the most relevant pages. For the *Parked Domain* category, we only analyzed the first 100 web pages per target trademark due to the enormous number of largely identical pages that we expect to accurately detect with a blacklist for unique HTML elements. We further analyzed every website in the *Affiliate Abuse* category to evaluate the performance of our approach.

We assigned one of the following threat levels to each relevant page in the *Unknown* and *Matching Images* category:

- **Suspicious:** Websites that most likely abuse the trademark.
- **Harmless:** Websites that are clearly related to the trademark but pose no threat. This includes sites that are owned by the target trademark owner and, for example, harmless fan sites.
- **Unrelated:** Websites that are unrelated to the target trademark.

We consider the distinction between the categories *Harmless* and *Unrelated* relevant since harmless websites, *e.g.*, other legitimate sites owned by the trademark, will often use logos and thus appear in the *Matching Images* category despite being harmless, while clearly unrelated pages will never use any of the trademark's logos.

Additionally, we assign one of the following four threats that we identified in the literature to each suspicious lookalike domain:

- **Fake Web Shop:** Websites that present a fake web shop. Some fake web shops never deliver any goods, whereas others deliver fake products.
- **Unsafe:** Web pages that contain malware, or unwanted software, or that try to perform social engineering attacks like phishing, which have been reported by Google's Safe Browsing technology employed in the Google Chrome Browser.[1]
- **Parked Domain** and Domain Selling: Domain names that were reserved for the sole purpose of generating ad revenue misusing the reach of the target trademark. Most of these domains can also be purchased with a high surcharge on the original price.
- **Other:** Websites with other malicious activities, *e.g.*, gambling or online dating sites that follow unfair business practices.

We further check whether the website contains a logo of the trademark to review the performance of our logo matching algorithm. These labels are assigned independently from the category. Therefore, it is possible to have a website with *Logo present* that remains in the *Unknown* category, *i.e.*, the logo matching has failed. Based on the images that contain logos and how they are present on a web page, we assign either one of these labels:

- **Standalone logo:** The website contains a trademark logo as a stand-alone image, *i.e.*, the logo is a distinct element of the web page and not embedded into any other image.

[1]`https://safebrowsing.google.com/`

- **Embedded logo:** The website contains a trademark logo as part of a larger image, *e.g.*, a product image.
- **No logo:** The website contains no trademark logo.

4.2 Provided Input Data

Our tool requires a TLD list, one or more target trademark domain names, and one or more target trademark logos.

4.2.1 TLD List

The list of domains must be as complete as possible to avoid false negatives in the results, *i.e.*, by skipping potentially cybersquatted domain names because the tool mistakenly assumes that they are inexistent as they do not exist in the TLD list. Therefore, we used a very comprehensive domain list available from *zonefiles.io*. This list is updated daily and contains over 200 million entries from about 1 500 TLDs.

4.2.2 Target Domains

In order to select target trademark domains which are valuable targets for attackers, we decided to use the top nine brands in the *luxury and apparel* category from the *Forbes* world's most valuable brands list that has been released in 2020.[2] We expect that these major companies have to constantly fight trademark abuse and therefore our investigations could lead to interesting discoveries, because data provided by the WIPO revealed that domain-squatting is a very prevalent issue for fashion trademarks.[3] For each of them, their main web presence was identified and that domain name was provided as target domain. The complete list of evaluated target domains can be seen in Table 4.1.

4.2.3 Target Trademark Logos

Initially, we planned to use logos from the provided target trademark domains for our analysis. However, we realized that lookalike domains were occasionally using a different logo that was still related to the brand, *e.g.*, a distinct logo used by the trademark owner to promote a certain collection of products like the "Nike Air Jordan" line of shoes. Consequently, we used the Google image search to identify the most relevant logos for each target target trademark. This task is not trivial: Google image search delivers thousand of samples for each target trademark, and we cannot know in advance the logos that are used on malicious sites. As search term we used the brand name represented by the target domain plus "logo," and we tried to identify about ten different logos for each brand, *e.g.*, variations with or without additional text or with text at different positions. We included them into our set of logos, because the current image matching algorithm cannot handle large visual differences between the original and the target logo.

[2]`https://www.forbes.com/the-worlds-most-valuable-brands/`

[3]`https://www.wipo.int/pressroom/en/articles/2019/article_0003.htmlhttps://www.wipo.int/export/sites/www/pressroom/en/documents/pr_2019_829_annex.pdf#annex4`

Target trademark	Target domain
Adidas	`https://www.adidas.com`
Cartier	`https://www.cartier.com`
Chanel	`https://www.chanel.com`
Gucci	`https://www.gucci.com`
Hermès	`https://www.hermes.com`
Louis Vuitton	`https://www.louisvuitton.com`
Nike	`https://www.nike.com`
Rolex	`https://www.chanel.com`
Uniqlo	`https://www.uniqlo.com`

Table 4.1: Evaluated target trademarks and their domains

4.3 Limitations

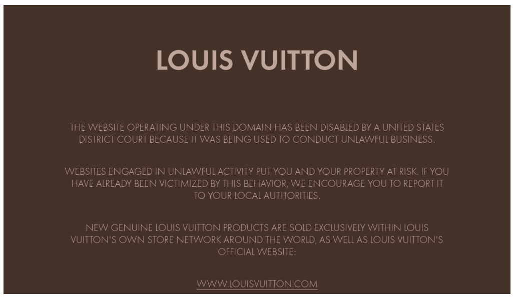
During our analysis, we identified two non-trivial problems that require a manual intervention, *i.e.*, the occurrence of common words within domain names and the domains reclaimed by the rightful owner of the embedded trademark.

4.3.1 Common Words

Some of our target trademarks share the name with other companies or individuals, however they usually operate in another business domain. There exist, for example, multiple people with the last name Cartier that are unrelated to the brand such as the corporate advisor Julie Cartier with her legitimate website `juliecartier.com`. Moreover, we found many more unrelated entries for Chanel, Gucci, Rolex, and Hermès. In particular, we could identify a large delivery company called *Hermes Europe GmbH* that introduced most of the lookalike domains for the target domain Hermès. In such cases, the domain name itself does not provide enough information to indicate trademark abuse. However, by considering the website content a reviewer can easily identify such sites and classify them as unrelated. Unfortunately, the automation of this process is a very challenging task.

4.3.2 Reclaimed Domains

Reclaimed domains are domains that have been acquired by a trademark owner to mitigate or prevent a potential trademark misuse. The companies behind the trademarks Chanel, Gucci, and Louis Vuitton aggressively reclaimed pages that have been abused by cybersquatters and replaced them with notification messages, *i.e.*, they usually tell a visitor that the previously hosted malicious website was taken down. In Figure 4.1 we show such notification messages for four of our target trademarks. Whereas the first and the second takedown notice from Louis Vuitton and Gucci, respectively, only provide basic information for the visitors, *i.e.*, the reason for the website takedown and a URL to their legitimate online shop, Chanel followed a more comprehensive approach: as shown in the third and fourth takedown notice they also


LOUIS VUITTON

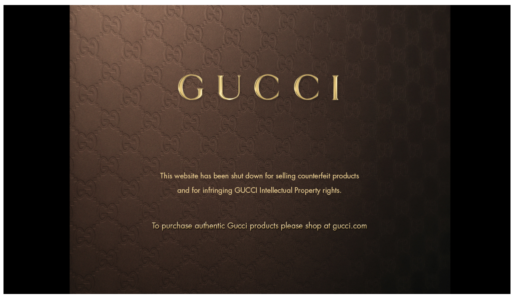
THE WEBSITE OPERATING UNDER THIS DOMAIN HAS BEEN DISABLED BY A UNITED STATES DISTRICT COURT BECAUSE IT WAS BEING USED TO CONDUCT UNLAWFUL BUSINESS.

WEBSITES ENGAGED IN UNLAWFUL ACTIVITY PUT YOU AND YOUR PROPERTY AT RISK. IF YOU HAVE ALREADY BEEN VICTIMIZED BY THIS BEHAVIOR, WE ENCOURAGE YOU TO REPORT IT TO YOUR LOCAL AUTHORITIES.

NEW GENUINE LOUIS VUITTON PRODUCTS ARE SOLD EXCLUSIVELY WITHIN LOUIS VUITTON'S OWN STORE NETWORK AROUND THE WORLD, AS WELL AS LOUIS VUITTON'S OFFICIAL WEBSITE:

WWW.LOUISVUITTON.COM

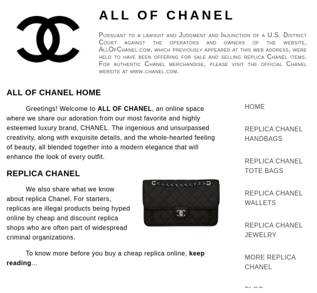
(a) Takedown notice from Louis Vuitton


GUCCI

This website has been shut down for selling counterfeit products and for infringing GUCCI Intellectual Property rights.

To purchase authentic Gucci products please shop at gucci.com

(b) Takedown notice from Gucci


ALL OF CHANEL

Pursuant to a lawsuit and Judgment and Injunction of a U.S. District Court against the operators and owners of the website, AllOfChanel.com, which previously appeared at this web address, were held to have been offering for sale and selling replica Chanel items. For authentic Chanel merchandise, please visit the official Chanel website at www.chanel.com.

ALL OF CHANEL HOME

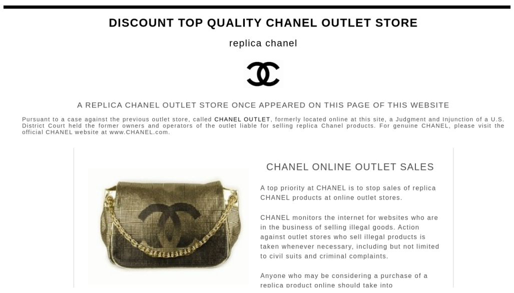
Greetings! Welcome to **ALL OF CHANEL**, an online space where we share our adoration from our most favorite and highly esteemed luxury brand, CHANEL. The ingenious and unsurpassed creativity, along with exquisite details, and the whole-hearted feeling of beauty, all blended together into a modern elegance that will enhance the look of every outfit.

REPLICA CHANEL

We also share what we know about replica Chanel. For starters, replicas are illegal products being hyped online by cheap and discount replica shops who are often part of widespread criminal organizations.

To know more before you buy a cheap replica online, **keep reading**...

(c) Takedown notice from Chanel


DISCOUNT TOP QUALITY CHANEL OUTLET STORE

replica chanel

A REPLICA CHANEL OUTLET STORE ONCE APPEARED ON THIS PAGE OF THIS WEBSITE

Pursuant to a case against the previous outlet store, called CHANEL OUTLET, formerly located online at this site, a Judgment and Injunction of a U.S. District Court held the former owners and operators of the outlet liable for selling replica Chanel products. For genuine CHANEL, please visit the official CHANEL website at www.CHANEL.com.

CHANEL ONLINE OUTLET SALES

A top priority at CHANEL is to stop sales of replica CHANEL products at online outlet stores.

CHANEL monitors the internet for websites who are in the business of selling illegal goods. Action against outlet stores who sell illegal products is taken whenever necessary, including but not limited to civil suits and criminal complaints.

Anyone who may be considering a purchase of a replica product online should take into

(d) Takedown notice from Chanel using a different design

Figure 4.1: Takedown notices from different corporations

Target trademark	Lookalikes	Inaccessible	Parked	Affiliate Abuse	Unknown	Matching Image
Adidas	4 251	2 265	554	5	1 291	136
Cartier	2 454	1 448	279	1	707	19
Chanel	6 593	3 281	994	2	2 302	14
Gucci	5 707	3 409	541	0	1 747	10
Hermès	8 819	3 198	891	1	4 711	18
Louis Vuitton	2 248	1 043	126	3	1 067	9
Nike	28 153	14 280	2 890	39	10 719	225
Rolex	16 630	7 218	652	2	8 673	85
Uniqlo	389	177	58	0	145	9

Table 4.2: Reported lookalike domains broken down by threats

explain visitors why they should avoid fake products, and they use different designs. We considered such reclaimed pages harmless, because the trademark owner controls them. The reclaimed domains account for most of the harmless websites that we found for each of our target trademarks. The automated identification of reclaimed domains is currently not supported by our tool.

4.4 Findings

The number of lookalikes found per trademark varied greatly as shown in Table 4.2. The first column denotes the target trademark, the second column the number of identified lookalike domains by our tool, and the third column the number of those domains that were inaccessible. The tool then categorized the lookalike domains that were accessible further into threat categories: column four presents the parked domains, column five the domains that are used for affiliate abuse, column six the domains that our tool could not precisely categorize, and the last column the domains whose websites contain images that matched at least one of the provided trademark logos.

According to the data, the prevalence of the threats seems to vary by trademark: whereas over 28 000 lookalike domains were reported for Nike, only 389 were found for Uniqlo. On average about 8 000 such domains were found for each trademark. We assume that Nike and Adidas are more popular targets for abuse, or they simply assign less resources to the website monitoring compared to the other brands based on their large number of lookalike domains. However, many lookalike domains could not be reached (48%), *i.e.*, the domains have been registered by someone, but were not in use. Less prevalent, but still omnipresent were the parked domains (10%). The number of domains that perform affiliate abuse was very small regardless of the trademark (< 1%). On the other hand, the tool reported many websites that it could not categorize on its own (42%), and only a minor subset of the websites contained images that our tool could match (< 1%).

Since our tool could not precisely determine the threat level of all identified lookalike domains, we manually assigned a threat level to the website of every domain in the *Unknown* and the *Matching Image* category that contained the target trademark in text form, *i.e.*, we had to exclude websites that did not

Target trademark	Total	Suspicious	Harmless	Unrelated
Adidas	377	307	52	18
Cartier	254	32	7	215
Chanel	472	78	169	225
Gucci	835	118	669	48
Hermès	2 105	102	271	1 732
Louis Vuitton	631	67	564	0
Nike	865	456	218	191
Rolex	410	261	84	65
Uniqlo	22	12	9	1

Table 4.3: Reported but obscure domains broken down by threat level

Target trademark	Total Suspicious	Miscellaneous	Fake Web Shop	Unsafe
Adidas	307	129	166	12
Cartier	32	20	10	2
Chanel	78	48	28	2
Gucci	118	53	53	12
Hermès	102	69	31	2
Louis Vuitton	67	44	22	1
Nike	456	201	246	9
Rolex	261	180	81	0
Uniqlo	12	8	2	2

Table 4.4: Number of squatted domains categorized by threat

mention the trademark to make the data feasible for a manual review. The number of websites in each threat level for every target trademark can be seen in Table 4.3. The first column shows the target trademark, the second column the total number of domains reported as *Unknown* or *Matching Image* that contain the trademark in the website content, the third column denotes the suspicious domains that require further investigations, the fourth column reports domains that are harmless for trademark owners, and the final column shows the number of unrelated domains. The number of suspicious domains for some trademarks is remarkable, however the numbers differ drastically between the different trademarks. The harmless domains are mostly caused by takedown notices. Gucci and Louis Vuitton seem to be very active in reclaiming websites, but Cartier and Uniqlo not at all. However, considering the enormous prevalence of lookalike domains for Adidas and Nike, their takedown engagement is clearly underwhelming. Unrelated domains were rather scarce except for Hermès: we found many websites that pointed to the "Hermes" delivery courier.

We continue our more detailed manual analysis only with the suspicious domains since these domains could become a threat for a trademark owner. Table 4.4 shows the number of suspicious pages that we found with the help of our tool, categorized by major threats. The *Total Suspicious* column contains all websites that used elements from the target trademark that are, as we expect, not officially sanctioned. The *Miscellaneous* column represents websites that offer dubious online gambling experiences, adult content, blogs with nonsensical content, a few parked domains that escaped detection in the prior steps, and other rather arbitrary content. The *Fake Web Shop* column presents the number of fake web shops that we have encountered. Finally, the *Unsafe* column shows the number of web pages that have been reported by Google's Safe Browsing technology, *e.g.*, available in the Google Chrome browser that indicates websites that contain malware, unwanted software, or that perform social engineering attacks such as phishing.

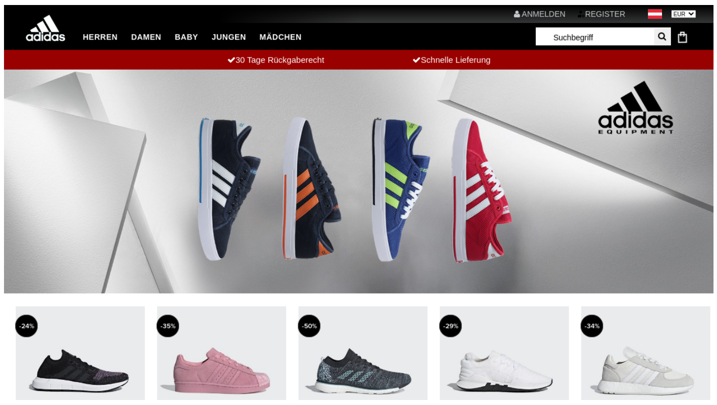
We could detect suspicious web shops for all evaluated target trademarks. The *Miscellaneous* and *Fake Web Shop* threats are common and account for 52% and 45% of the suspicious websites, respectively.

In general, we could find a plethora of different abuses for each trademark, which contributed to the high number of sites labelled as *Miscellaneous*. Some of these websites presented advertisements or links to other illicit sites like web shops for replicas. Moreover, an abundance of blog like websites with sometimes nonsensical content was hosted under squatted domain names. We assume that such sites exist to gather ad revenue and to preserve the domain for a different purpose in the future, *e.g.*, to run a phishing campaign. Three examples from the *Miscellaneous* category are shown in Figure 4.3. The first example shows a blog with links to various gambling sites that abuses the Louis Vuitton trademark to gain reach. The second example shows a valuables buyer that potentially misuses the Rolex trademark both in the domain name (`gold-diamond-rolexbuyers.com`) and the content of the site. Finally, the last example presents satirical content that misuses the Nike trademark.

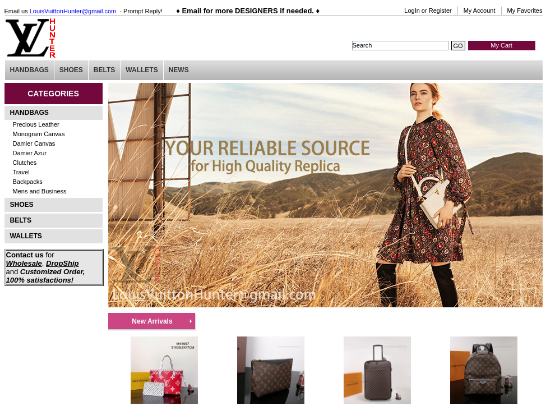
Moreover, trademark abuse through fake web shops is very prevalent. A selection of different fake web shops can be seen in Figure 4.2. The first fake web shop is specific to Austria and imitates the official design. Although the translations are complete they are rather poor in quality. The second fake web shop advertises Louis Vuitton replicas and announces large discounts unlike official shops. As we frequently observed, the listed product ranges are very comprehensive to convince a potential buyer that the shop maintains a large stock of different products. The last fake shop advertises Rolex watch replicas and even provides an online chat functionality to connect with a sales representative. Compared to the other threats, we could only find a few unsafe pages. It appears that squatted domains are rarely used for phishing and malware distribution, at least for the selected trademarks.

4.5 Tool Evaluation

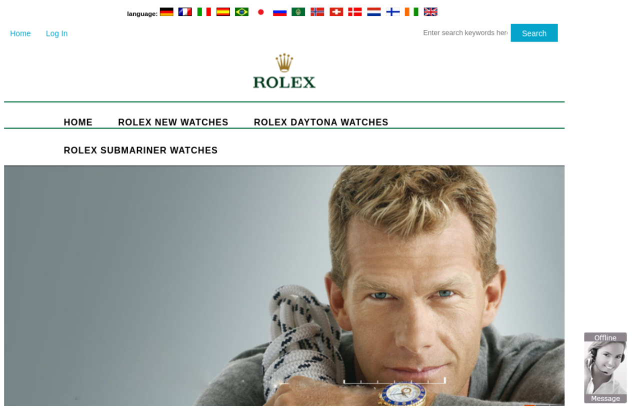
Based on the reported lookalike domains from our tool, we investigate the quality of the used logo matching, and how well our tool recognized parked domains and affiliate abuse.



(a) Fake web shop for Adidas that targets Austrian customers

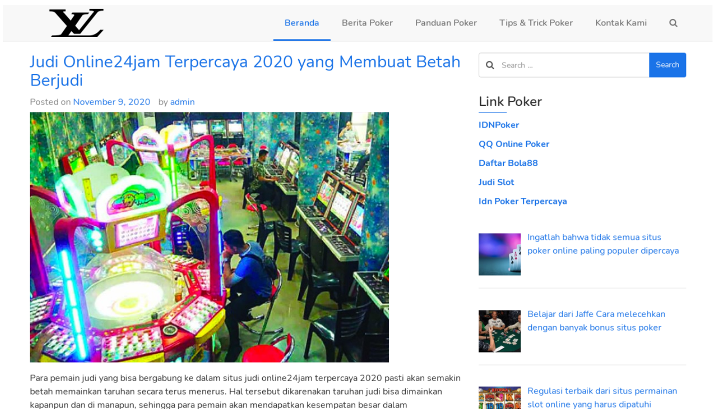


(b) Fake web shop for Louis Vuitton replicas that offers a large variety of items



(c) Fake web shop for Rolex replicas with an online chat functionality

Figure 4.2: Diversity of fake web shops



Judi Online24jam Terpercaya 2020 yang Membuat Betah Berjudi

Posted on November 9, 2020 by admin

Para pemain judi yang bisa bergabung ke dalam situs judi online24jam terpercaya 2020 pasti akan semakin betah memainkan taruhan secara terus menerus. Hal tersebut dikarenakan taruhan judi bisa dimainkan kapanpun dan di manapun, sehingga para pemain akan mendapatkan kesempatan besar dalam

Link Poker

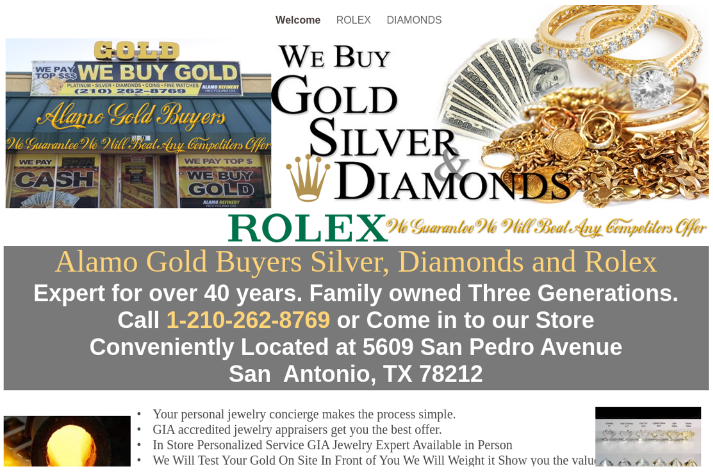
IDNPoker
QQ Online Poker
Daftar Bola88
Judi Slot
Idn Poker Terpercaya

Ingatlah bahwa tidak semua situs poker online paling populer dipercaya

Belajar dari Jaffe Cara melecehkan dengan banyak bonus situs poker

Regulasi terbaik dari situs permainan slot online yang harus dipatuhi

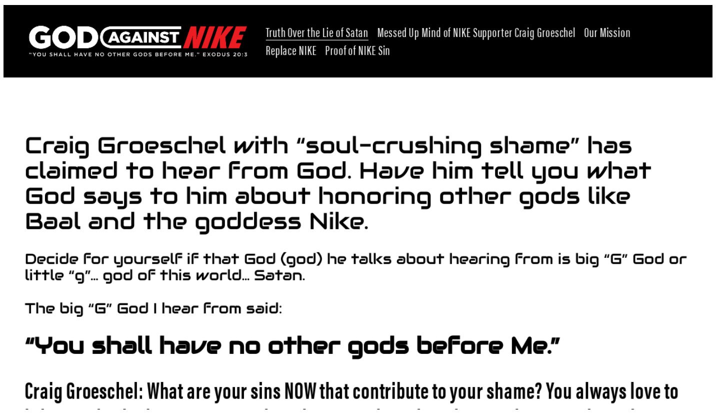
(a) Misuse of a trademark's reach



Alamo Gold Buyers Silver, Diamonds and Rolex
Expert for over 40 years. Family owned Three Generations.
Call 1-210-262-8769 or Come in to our Store
Conveniently Located at 5609 San Pedro Avenue
San Antonio, TX 78212

- Your personal jewelry concierge makes the process simple.
- GIA accredited jewelry appraisers get you the best offer.
- In Store Personalized Service GIA Jewelry Expert Available in Person
- We Will Test Your Gold On Site In Front of You We Will Weight it Show you the valu

(b) Unclear trademark affiliation


GOD AGAINST NIKE
"YOU SHALL HAVE NO OTHER GODS BEFORE ME." EXODUS 20:3
Truth Over the Lie of Satan
Messed Up Mind of NIKE Supporter Craig Groeschel
Our Mission
Replace NIKE
Proof of NIKE Sin

Craig Groeschel with "soul-crushing shame" has claimed to hear from God. Have him tell you what God says to him about honoring other gods like Baal and the goddess Nike.

Decide for yourself if that God (god) he talks about hearing from is big "G" God or little "g"... god of this world... Satan.

The big "G" God I hear from said:

"You shall have no other gods before Me."

Craig Groeschel: What are your sins NOW that contribute to your shame? You always love to

(c) Unfavorable content that relates to a trademark

Figure 4.3: Typical threats observed in lookalike websites

Target trademark	TP	FP	FN	TN	Recall	Precision
Adidas	110	15	197	55	0.358	0.880
Cartier	4	5	28	217	0.125	0.444
Chanel	9	5	69	389	0.115	0.643
Gucci	4	3	114	714	0.034	0.571
Hermès	1	7	101	1996	0.010	0.125
Louis Vuitton	6	2	61	562	0.090	0.750
Nike	133	39	323	370	0.292	0.773
Rolex	41	12	220	137	0.157	0.774
Uniqlo	2	4	10	6	0.167	0.333

Table 4.5: Confusion matrix of the image matching for each trademark

4.5.1 Image Matching

Image matching is a key technique to detect malicious lookalike domains which pose a clear threat to the trademark. Moreover, it can reduce the number of irrelevant results from the usually large list of reported lookalikes to ease a subsequent manual analysis. In the following, we evaluate the quality of our logo matching, and we investigate whether that feature is useful to detect highly problematic web pages.

Table 4.5 shows a confusion matrix of the image matching for each trademark after a manual review. We used the obscure domains from Table 4.3 as dataset, because these websites are of inherent interest for trademark owners due to their unclear affiliations. The first column represents the target trademark, and the second to the fifth column present the true positives, the false positives, the false negatives, and the true negatives regarding the existence of a target trademark's logo. Finally, the remaining two columns reveal the recall and the precision based on the values from the left. We can see that for most trademarks very few websites contained an image that our tool could match against one in our provided set of logo images. However, the logo detection apparently works better for both Nike and Adidas with each of them achieving more than 100 matches. Nevertheless, the image detection achieved a rather high precision (average: 0.59), but a very low recall (average: 0.15). The reason for the low recall is primarily the large number of false negatives that our image detection algorithm reported. Therefore, the used logo matching strategy is not satisfying for the identification of most trademark logos.

We are interested in the reasons for the rather low detection quality of our image matching technique. Therefore, we reviewed each logo in the suspicious sites that we identified in Table 4.4 one more time to further reason about the culprits of our implementation. For this evaluation we use the suspicious sites that contain a trademark logo, *i.e.*, a subset of the obscure domains that frequently misuse image resources from the target trademark's original website. We define a standalone logo as a distinct element of the web page representing merely the logo *i.e.*, an image or svg element, and an embedded logo as a trademark logo as part of a larger image, e.g., a product photo. Table 4.6 shows how many suspicious web pages use a logo from the original trademark, and how the logo has been added to the page. The first column denotes the

Target trademark	Suspicious	Standalone logo	Standalone or Embedded Logo	Difference
Adidas	307	141 (46%)	173 (56%)	+10%
Cartier	32	13 (41%)	15 (47%)	+6%
Chanel	78	22 (28%)	34 (44%)	+16%
Gucci	118	31 (26%)	44 (37%)	+12%
Hermès	102	27 (26%)	38 (37%)	+11%
Louis Vuitton	67	14 (21%)	24 (36%)	+15%
Nike	456	186 (41%)	282 (62%)	+21%
Rolex	261	44 (17%)	116 (44%)	+27%
Uniqlo	12	2 (17%)	3 (25%)	+8%

Table 4.6: The usage of trademark logos

Target trademark	customized logos	using altered text	using altered logo	using alternative logo
Adidas	85	11	32	42
Cartier	4	2	2	0
Chanel	15	9	5	1
Gucci	20	9	8	3
Hermès	21	9	10	2
Louis Vuitton	12	6	4	2
Nike	123	72	27	24
Rolex	20	11	7	2
Uniqlo	1	1	0	0

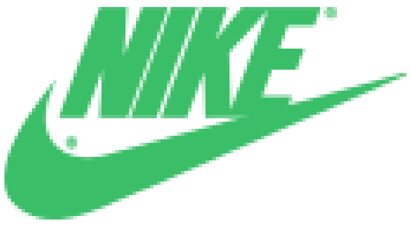
Table 4.7: Number of suspicious websites with altered or alternative logos

target trademark, the second column the total number of suspicious websites, the third column the number of websites that contain at least one standalone logo, *i.e.*, a logo in an image without additional content and their fraction of the second column, the fourth column reports the same for websites that contain at least one logo regardless of how it was used, *e.g.*, the site might contain the logo only in a product image, and finally, the last column shows the difference of the reported fractions. If we could accurately detect standalone images we would on average detect logos in about 29% of all suspicious websites, and that value would further increase on average by 14% if we could also accurately detect embedded logos. In other words, an average recall of about 0.29 (only standalone logos) or 0.44 (standalone and embedded logos) is achievable using a more sophisticated logo detection algorithm.

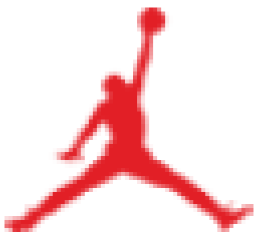
As we found, a trademark logo in a standalone logo is more often altered than in a logo embedded in another image. Therefore, we are interested in these changes to understand the requirements for a well-performing matching algorithm. A common change that we observed was the placement of additional text around the logo and the consolidation with other logos. Further changes include the scaling, the placement, and the used color palette, however we have not encountered any image mirroring or rotation. A selection of altered logos that most presumably try to mislead image detection tools can be found in Figure 4.4. The

(a) Adidas logo that uses extra white space on the left

(b) Adidas logo that uses an entirely different background



(c) Nike logo that uses a different color



(d) Nike product logo that uses a different color

Figure 4.4: Altered trademark logos found on lookalike domains

Target trademark	Reported domains	Missed parked domains
Adidas	377	29 (8%)
Cartier	254	12 (5%)
Chanel	472	15 (3%)
Gucci	835	19 (2%)
Hermès	2105	26 (1%)
Louis Vuitton	631	3 (0.5%)
Nike	865	29 (3%)
Rolex	410	59 (14%)
Uniqlo	22	3 (14%)

Table 4.8: Assessment of our parked domain removal strategy

first logo has been shifted to the right, the second logo leverages an uncommon image underlay, whereas the third logo uses a different color and has been shifted to the left. The last logo only uses a different color scheme. We quantified the use of such changes in Table 4.7. The first column lists the target trademarks and the second column refers to the number of logos that we found on suspicious web pages that were not present on the trademark's home page, ignoring simple format conversion *e.g.*, from `.svg` to `.jpg` and added background. The remaining three columns describe the changes that have been applied to the logo, *i.e.*, *using altered text* reports how many of these logos underwent textual changes, *i.e.*, most notably the addition of text, *using altered logo* reports how many of these logos underwent minor image manipulations but still reveal the affiliation with an image from the trademark owner, and finally, in *using alternative logo* we present the number of logos that were entirely different from the ones currently used by the trademark owner on their official website, *e.g.*, outdated logos that are no longer used by the trademark owner or the use of product specific logos to represent the entire trademark. In cases where a website could be assigned to both categories, *i.e.*, *using altered text* and *using altered logo* we assigned it to the *using altered text* category, because the image hash difference introduced by added text usually outweighs any difference introduced by minor image changes. A significant proportion of all standalone logos has been altered from all trademarks. Most common was the change of text, *e.g.*, the insertion or removal of the trademark name near the trademark logo. Less common was the use of an entirely different logo, *e.g.*, the use of a product logo ("Air Jordan") instead of the regular trademark logo ("Nike"). Most image changes were rather simple, *i.e.*, a color change or inversion, or simple modifications of the logo's outline.

4.5.2 Parked Domain Detection

Filtering parked domains can drastically reduce the number of web pages that require a manual intervention. To assess the quality of our parked domain detection, we refer to the parked domains that exist in the dataset that we used for our manual analysis, because it should not anymore contain parked domains as they should have been removed in the process.

Table 4.8 presents the results. The first column denotes the target trademark and the second column the number of reported websites in our obscure dataset. The third column presents the number of the missed parked domains and their fraction, respectively. We can see that the fraction of missed parked domains varies from less than 1% to 14%. In other words, there exist still some parked websites in the result set that must be removed manually. Additionally, we randomly sampled 100 domains from each trademark that have been labelled as *Parked Domain*, except for Uniqlo where we only had 58 parked domains to begin with, and checked whether they are indeed parked domains. We could not find any non-parked domain in this new randomly sampled dataset that we used, *i.e.*, every randomly sampled domain has been correctly labelled as *Parked Domain*. Therefore, we expect a rather low rate of false positives for our parked domain detection strategy. In summary, although the precision is very high the blacklist used by the algorithm should be expanded to accommodate for more kinds of parked domains.

4.5.3 Affiliate Abuse

For each trademark we verified every web page that has been flagged by our tool as affiliate abuse, since there exist only very few of them and it is a very time consuming task to find them manually. In fact, we did not detect any additional affiliate abuse sites during our manual investigations. We found that every flagged web page was indeed performing some kind of affiliate marketing. This indicates that our tool can accurately detect affiliate abuse, despite the small list of unique query values and parameters that are used for the detection.

4.6 Discussion

Logos are a useful feature to identify the most relevant lookalike domains at least for those trademarks that particularly suffer from many fake web shops. However, the image detection algorithm provides room for improvements: typical logo modifications like adding custom text or embedding a logo into a product image irritate the used image hashing function. In practice, although the image matching algorithm performs a fuzzy matching, extra text frequently biased the hash and thus allowed the logo to escape from our detection. Nevertheless, the scale and color variations were usually matched by our detection algorithm. In order to achieve a reasonable performance, we had to provide for each trademark multiple different logos and variations of them, *i.e.*, with and without additional text. Moreover, the explicit focus on standalone logos was unfortunate. Although standalone logos are the most prevalent kind of logos in malicious websites, many malicious web shops embed logos in product images that we missed. Overall, our approach reached an average precision of 59%, *i.e.*, from all manually analyzed web pages where the algorithm detected a logo the majority were most likely malicious.

Despite the underwhelming performance of the logo matching, the detection for both *Affiliate Abuse* and *Parked Domain* was reliable and accurate. Removing these threat types from the list of reported domains drastically reduces the number of websites that require a manual intervention. We cannot provide a reliable estimation of the recall for any of these two categories since we were unable to manually analyze every lookalike domain due to time constraints. However, given the low number of parked domains that were left in the obscure dataset we can assume a high recall. For both *Affiliate Abuse* and *Parked Domain* we found no false positives, *i.e.*, all labeled pages that we looked at were abusive according to our definitions. This implies that our blacklist approaches for both of these categories also yield a high precision. However, these results should be taken with a grain of salt, because we cannot know the intention behind each domain registration, and thus some of them might actually not be performing cybersquatting.

Our tool was able to identify a considerable number of fake web shops for both the Nike and the Adidas trademark, which we consider a serious threat for them. Ultimately, we provided twelve domains for which we were certain that they are malicious to Nike's trademark abuse team for review. However, we did not receive any written or verbal notification from Nike. Nevertheless, of the twelve domains reported at 08-MAR-2021, six were still active a month later, and as of 17-JUL-2021 only four domains still remain online.

5 Threats to Validity

The applicability of our results might be limited. In our case study we focused on nine well known brands from the same category, *i.e.*, luxury and apparel, to ensure that a manual analysis of the reported web pages will be feasible. Therefore, the result of our case study might not be fully applicable for trademarks in other business categories. This is an inherent threat, because our analysis is only about the prevalence and use of lookalike websites targeting corporations that produce goods for consumers, which may not be generalized.

We introduced arbitrary decisions during the implementation and evaluation of our work. During the manual analysis we only analyzed web pages in the *Unknown* category that contained the trademark in their content to reduce the required manual effort for the analysis. There is a chance that we missed malicious pages. However, we consider that a non-issue since most squatted domains try to convince the visitors that they are related to the target trademark, which we consider difficult without mentioning trademark in textual form. Another arbitrary decision was the selection of logos that we used to detect malicious web pages. However, this is an inherent limitation of our tool and its users should know the relevant logos for the domains they want to protect. To overcome this threat, we tried to collect as many sufficiently different logo images as possible from Google's image search for each trademark.

There is a threat to correctness through a lack of information regarding the ownership of domains. Given the large number of manually analyzed websites it was very challenging to identify the appropriate threat level for each entry, especially for the distinction between harmless and suspicious. Additionally, fake web shops were often surprisingly well made and we had to manually browse through their content to find hints for malicious behaviour. However, we did not issue any test orders on them. To mitigate this threat, we strove to find information that revealed their true identity. We considered a domain illicit, for example, when contact information was missing or the contact possibilities were only realized as a simple web form without any phone numbers or address information.

We do not know the exact affiliate marketing terms for each trademark. We cannot definitely say whether a web page is performing an affiliate abuse purely based on the fact that it is participating in an affiliate program using a squatted domain. Some trademarks, for example, could consider typo-squatted domains that redirect to their own home page a legitimate form of affiliate marketing as discussed in the study of Amarasekara *et al.* [4]. In general, without studying and understanding the exact terms of affiliate marketing for each target trademark we cannot decide whether a squatted domain is performing a fraud.

Domain similarity is highly context sensitive. From a security perspective, there exist no commonly agreed criteria on whether a domain name is too similar to another to represent a threat. Usually, the content of a web page must be examined to provide an informed decision. Unfortunately, parked domains in particular host usually no specific content that reveals their target. Eventually, deciding whether a specific domain name is a case of domain-squatting is often a legal matter since the intention behind a domain registration is often only known by the registrant. Since we cannot entirely mitigate this threat we assumed that all parked domains are indeed targeting the respective trademark, and we used, where applicable, the recommended algorithm parameters.

6 Conclusion

In this thesis, we investigated cybersquatted domains. We cover how they can be detected, and explained why it is difficult to decide whether a lookalike domain is malicious or not. Logo matching is presented as a new feature that could be used to automatically separate harmless from malicious lookalike domains. Consequently, we developed a tool that is able to detect squatted domains on a global scale and highlight problematic domains with the help of logo matching implemented using image hashing. To validate our tool, we conducted a case study were we analyzed the lookalike domains and the content hosted on these domains for nine global trademarks in the apparel and luxury sector. We performed a manual evaluation on more than 5 000 lookalike domains reported by the tool and identified 1 433 pages that host malicious content. The case study showed that such a tool could be useful especially for the detection of fake web shops since we could find 639 suspicious web shops that targeted well known trademarks like *Nike* and *Adidas*, despite the measures taken by the trademark owners to protect their intellectual property. However, the proposed image hashing algorithm was a bad fit, since common logo manipulations were causing large differences in the similarity score and image hashing cannot be applied to logos embedded into other images like product photos which occasionally contained the only occurrence of the logo. In summary, we found that trademark logos are a useful feature to detect cybersquatted domains and that our tool can successfully detect and report malicious domains for every trademark that we analyzed in our study.

Fair Use

Bibliography

[1] S. Abdelnabi, K. Krombholz, and M. Fritz. VisualPhishNet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1681–1698, 2020.

[2] S. Afroz and R. Greenstadt. PhishZoo: Detecting phishing websites by looking at them. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 368–375. IEEE, 2011.

[3] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*. Internet Society, 2015.

[4] B. Amarasekara and A. Mathrani. Exploring risk and fraud scenarios in affiliate marketing technologies from the advertisers perspective. *arXiv preprint arXiv:1606.01428*, 2016.

[5] A. S. Bozkir and M. Aydos. LogoSENSE: A companion hog based logo detection scheme for phishing web page and e-mail brand recognition. *Computers & Security*, 95:101855, 2020.

[6] N. Chachra, S. Savage, and G. M. Voelker. Affiliate crookies: Characterizing affiliate marketing abuse. In *Proceedings of the 2015 Internet Measurement Conference*, pages 41–47, 2015.

[7] P. Gill and R. Nithyanand. Extortion or expansion? An investigation into the costs and consequences of ICANN's gTLD experiments. In *Passive and Active Measurement: 21st International Conference, PAM 2020, Eugene, Oregon, USA, March 30-31, 2020, Proceedings*, volume 12048, page 141. Springer Nature, 2020.

[8] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 569–586, 2017.

[9] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe. Enabling network security through active DNS datasets. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 188–208. Springer, 2016.

[10] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool. WebVision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] P. Lv, J. Ya, T. Liu, J. Shi, B. Fang, and Z. Gu. You have more abbreviations than you know: A study of abbrevsquatting abuse. In *International Conference on Computational Science*, pages 221–233. Springer, 2018.

[13] T. Moore and B. Edelman. Measuring the perpetrators and funders of typosquatting. In *International Conference on Financial Cryptography and Data Security*, pages 175–191. Springer, 2010.

[14] N. Nikiforakis, S. Van Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen. Bitsquatting: Exploiting bit-flips for fun, or profit? In *Proceedings of the 22nd International Conference on World Wide Web*, pages 989–998, 2013.

[15] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[16] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool. LLD - Large Logo Dataset, version 0.1. `https://data.vision.ee.ethz.ch/cvl/lld`, 2017.

[17] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang. Needle in a haystack: Tracking down elite phishing domains in the wild. In *Proceedings of the Internet Measurement Conference 2018*, pages 429–442, 2018.

[18] G. Wang, H. Liu, S. Becerra, K. Wang, S. J. Belongie, H. Shacham, and S. Savage. *Verilogo: Proactive phishing detection via logo recognition*. Department of Computer Science and Engineering, University of California . . . , 2011.

[19] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels. Strider Typo-Patrol: Discovery and analysis of systematic typo-squatting. *SRUTI*, 6(31-36):2–2, 2006.

[20] Y. Zeng, T. Zang, Y. Zhang, X. Chen, and Y. Wang. A comprehensive measurement study of domain-squatting abuse. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[21] J. Zhao, R. Masood, and S. Seneviratne. A review of computer vision methods in network security. *arXiv preprint arXiv:2005.03318*, 2020.