(Aster)-picking through the pieces of short URL services

An investigation into the maliciousness of short URLs

Peter Boers
`peter.boers@os3.nl`

Robert Diepeveen
`robert.diepeveen@os3.nl`

February 7, 2016

Abstract

This study uses a brute-force approach to check a sample of the total keyspace of three short URL service providers for malicious links. The services that were examined are `bitly`, `TinyURL` and `goo.gl`. It appears that every service has been used for shortening malicious links at some point in time in the past. After collecting information about 1.4 million links, a total of 948 malicious links were found that are unmarked by the service providers. During the experiment a link was marked malicious if it appeared in one of the following blacklists: SpamHaus domain blocklist, SpamHaus ZEN blocklist, Google Safe Browsing and PhishTank. Incorporating the number of links marked as malicious by the services themselves we found that `TinyURL` contained 5.17% malware, `bitly` contained 0.05% malware and `goo.gl` contained 0.01% malware. The study then looked into a malicious website which was present in about 50% of the undetected malware hits called `asterpix.com`. This site manifested itself as an entry point to a sophisticated network that targets visitors on the basis of their location.

1 Introduction

Short URLs have infiltrated our daily lives in such a way that we cannot think of services such as Twitter and Facebook without them. In this day and age of social media, sharing and mobile Internet usage, people are spending more and more time looking at very small screens [1]. When travelling on public transport or just walking around in towns and cities, one notices people staring at their mobile devices in constant communication with their friends, family and colleagues through online social media.

Short URLs prettify and compress the size of longer and less readable links. Usage of these services such as `bit.ly` exploded when they were made Twitters standard URL shortening service in 2009 [26]. Short URLs enable users to share media and links with their followers in such a way that it fits in a tweet[1].

When looking at the `alexa.com` rankings of high traffic sites, services like `bit.ly`, `goo.gl`, `tinyurl.com` and `t.co` have featured in the top 1000 sites over the past three months[2]. The short URL services are now used in a much broader sense. It is more and more common to find them in emails and/or messaging services as they enable users to make their text shorter.

Because short URLs are effective in hiding the real content of a URL creators of malware, phishing sites, spammers and scammers now have a means to obfuscate and hide their long URLs. The usage of short URLs in malicious campaigns has been investigated as it spreads through Twitter [18, 15, 16, 7]. However, there has been little research into how much malicious redirections are being done with short URLs and where these redirections lead.

Upon finding a number of malicious sites we choose to examine the active malware site `asterpix.com`[3] and attempt to find out how the locality of a user influences the redirect graph of this site. Previous research [6, 19, 16] has shown that JavaScript, IFrames and meta tags have been analyzed to examine how these influence what a user sees, but what about locality? Location aware online advertisement campaigns are abundant, but how does a malware network make use of this information? `asterpix.com` shows that adware and malware networks are becoming more and more sophisticated in how they redirect their visitors to ensure their effectiveness and at the same time mitigate their detection.

What are short URLs?
Below you find a demonstration of how a short URL service works. Assume you would like to compress

[1]140 Characters
[2]`alexa.com` - January 2016
[3]Active as of January 2016

the following URL:

`http://example.com/xxx/yyy/zzz/111/222/333`

After compression by a short URL service it will resemble the format below, thus creating the *short URL*:

`http://exmpl.cm/ABc123`

Users can generate the short URLs by visiting the service's site and filling in a long URL they want to compress. The service will then return a persistent, shortened URL that will always be associated with the original long URL. How these links are built up is discussed in Section 5.

Once in possession of the short URL, users can spread this to their network. When clicking on the short URL, visitors are permanently redirected[4] to the long URL. In some cases the short URL service providers make statistics available about the links such as referrals and clicks.

Contributions
This research contributes in the following ways:

- It takes a snapshot of some well known short URL providers and measures the effectiveness of their abuse filtering mechanisms.
- To our knowledge it presents the first observation of a malware network that evaluates user on their geolocation and attempts find weaknesses in this network to help services such as Google Safe Browse effectively warn visitors.
- It suggests methods which after further research may help frustrate the efforts of locality based malware networks.

2 Research Questions

As stated in the introduction this research investigates questions related to abuse of short URLs. It does this by collecting a uniform dataset of URLs from a number of short URL providers. In light of what can be interpreted from the data, this paper answers the following question:

What portion of the short URL services are used for malicious purposes and what does the abuse look like?

Below a number of subquestions are defined to help focus the research:

- Which service provides proportionally the most short URLs flagged as malicious?
- What properties can be observed in encountered malicious sites?

[4]`301` Redirect

3 Definitions

Spam is defined as mass online sending of unsolicited messages, often containing links to phishing sites and/or adware. The act of spreading these messages is called **spamming**.
Phishing is the act of imitating a web-page to illegally retrieve credentials from gullible users. Phishing campaigns are often spread through spamming through e-mail or on Social Media.
A **URL** is defined in RFC3986 [5]. For this research it is important to know that a URL consists of a schema, a domain and a path. The schema indicates which protocol the requester should use to request the resource, the domain (which could be an IP Address) points to the location where the resource can be found and the path indicates the the server which resource it should return. A URL may look like: `scheme://domain/path`. The usage of "URL" and "link" in this paper is interchangeable.
Short URLs point to a long URL[5]. They leverage certain HTTP status codes to instruct the user's browser to redirect to the destination page. The actual status codes are described in RFC 2616 [10]. The path of the short URL[6] is called the **(short) hash** of the short URL. The collection of all possible hashes is called the **key space**.
A **temporary redirect** contains a response with the status code 302, whereas a **permanent redirect** is a 301 response. Short URLs pointing to long URLs often redirect visitors with a permanent redirect when the URL is not flagged as malicious.
Malicious URLs are URLs which point, redirect or serve pages which can be classified as serving malware, adware or phishing sites; or are URLs which point to IP addresses from which spam campaigns were ran. IP addresses can also be malicious URLs. In this research we treat a link as malicious if either the short URL service provider or one of the blacklists mentioned in Section 6.2 marks it as malicious.
We distinguish between **detected** and **undetected** malicious URLs. Detected malicious URLs indicate the the short URL service has picked up on the malicious nature of the long URL and points to a landing page, whereas undetected malicious URLs are URLs which have not been picked up by the short URL service.
Link rot is the fact that hyperlinks, over time, may go offline or point to inactive pages.

[5]Note that 'long' URLs are not by definition longer than short URLs
[6]The portion after the domain name

4 Related Work

The following related work helps place this research within the scope of the field. A number of studies are highlighted that influence the direction and nature of this paper while answering the research question.

4.1 Anatomy of abuse with short URLs

Previous work has identified that there are on average 777 new phishing attacks using urls a day with about 5037 page views [11]. Phishing URLs are relatively short lived but have around a 9% success rate [11].

Attempts have been made by Chhabra et al to use public information from PhishTank [7] in order to classify long URLs as malicious. The authors collected a number of `bitly` links and tested them against the PhishTank database. Chhabra et al mainly focused on the spreading of malicious URLs through the Twitter network. The authors found that Twitter accounted for 23% of all referrals to phishing URLs as opposed to 12% in general URL traffic.

The 'WarningBird' framework [16] was created as a method to classify the safety of URLs. This work once again focused on Twitter. The authors identified that (some) phishing URLs made use of "conditional redirections", based on the `User Agent` of a visitor. The WarningBird framework is able to classify URLs as malicious which make use of conditional re-directions, as opposed to Google Safe Browse.

In light of the previously mentioned results, Google has created its Safe Browsing Service [25]. Google aims to improve on the speed and accuracy of PhishTank and the now no longer active service of WarningBird [16], by incorporating the ideas of those services into their own. Whittaker et al [25] claims that the Google Safe Browsing API has a 0.1% false positive rate, 90% true positive rate and that they are able to reduce the detection time to hours instead of days compared to WarningBird, thereby providing a means to accurately determine if a URL is hosting malicious content. Furthermore Google actively crawls sites and does not rely on user feedback like PhishTank does.

By combining the Safe Browsing API, WarningBird's [16] previous work on the anatomy of Twitters short URL malware attacks, the PhishTank database and this research it is possible to gain insight into the general anatomy of short URL abuse on multiple services.

As well as using Google's Safe Browsing service we propose to use a more traditional Domain Name System Blacklist (DNSBL) as introduced by the Internet Engineering Task Force (IETF) [17]. SpamHaus is a well known DNSBL and we will use this service next to the Safe Browsing service to see if these more traditional blacklists also flag malicious short URL sites. Recent studies have indicated that they are not as effective as Google as they have a false positive rate of up to 35% [21].

4.2 Short URLs in the wild

Maggi et al [18] researched the countermeasures deployed by short URL services. The authors conclude that there was a relatively small number of malicious short URLs in the wild. The method of Maggi et al [18] has some similarities to this work, however the authors collected data through a browser plugin in which users were asked to flag URLs they deemed malicious. The researchers then analysed and checked each URL users entered into the database. It greatly depended on how, when and which URLs users submitted for the researchers to check them.

Klien et al[15] previously investigated the shortening service `qr.cx`. In their study the researchers observed that of all links in their sample, about 80,24% were spam and that the creation ratio was highest in South America and Africa, whereas the USA resolved most short URLs. Both of these papers [18, 15] however use data that is up to 4 years old. Antoniades et al [2] indicate that even though 50% of short URLs are not ephemeral, after 1 month about 85% are inactive and after about 2 years $\geq$ 99% are inactive. This would indicate that new research is likely to achieve a new set of active short URLs, compared to the older research.

4.3 Javascript Redirection, URL obfuscation and IFrames

How malware sites obfuscate and redirect vistors has been studied a number of times in the past. For the categorisation and analysis of `asterpix.com` we rely on the work from Chellapilla et al [6]. The authors describe a number of common methods used by malicious sites with JavaScript to redirect their visitors to redirect chain endpoints. Chellapilla et al [6] found that malicious sites use a number strategies to obfuscate redirection; by splitting the URL in code or by using methods such as `eval` execute obfuscated code. A study by Provos et al[19], shows how IFrames can be used next to JavaScript as a method to let browsers download programs or redirect to certain endpoints in malware networks. These IFrames are loaded with no pixels or they are transparent in the browser, allowing malicious code to be executed. Finally in their study Garera et al [11] analyze how URLs can be obfuscated to mislead visitors through squatting and how the URL path can be used to mislead gullible visitors in thinking that they are visiting a legitimate site.

The frameworks and methods described in these papers help us to investigate and follow the redirect chains of any encountered malware site to help classify and categorize them.

5 Short URL providers

In previous work [2, 15, 16] the focus of the researchers was mainly on `bitly` and `t.co`. These papers look at how short URLs have been used in spam, scam, and phishing campaigns on Twitter as described in Section 3. `bitly` in particular received much scrutiny after Twitter chose it to be its standard URL shortening service. When Twitter launched its own service, `t.co`, focus shifted to this as it took over the short URL function of `bitly` in tweets.

The two year long study conducted by Maggi et al [18], discovered that the most common used short URL services in descending order are; `bitly`, `t.co`, `TinyURL` and `goo.gl`.

5.1 The providers

This research chooses to examine `bitly`, `TinyURL` and `goo.gl`. Even though `t.co` is the most visited shortening service website according to `Alexa.com`[7], we have decided not to investigate it. As Section 4.1 states, the manifestation of spam and malicious short URLs has been intensively scrutinised. Furthermore, Twitter automatically inputs all "tweeted" links in their shortening service [24]. Twitter users might also use short URL services like `bitly`, `TinyURL` and `goo.gl` in their tweets, causing double work and skewed results if `t.co` would be investigated.

5.2 Size of the key space

`TinyURL` states on its homepage[8]: ***Making over a billion long URLs usable!*** It is unknown how providers assign short hashes to long URLs. However, to achieve a uniform sample the whole available key space needs to be searched. This section explains how the key space of the providers is built up. Section 6.1 will explain how we propose to search the key space in a uniform manner to analyse a representative subset. Please note that the short URL providers do not take up the whole key space. This means that only a (small) portion of hashes actually point to a long URL.

A hash is a string of numbers and letters. Because every integer can be expressed in any base, any random integer can be expressed in a base such that possible letters and numbers are equal to the way a hash is built up. This aids in generating the number of hashes.

Bit.ly uses a 7-digit hash as of January 2016. Earlier results [18, 2, 22] indicate that `bitly` uses a 6-digit hash. In this hash all alphanumeric characters may occur. This makes the hashes of `bitly` numbers in base-62. The total key space is shown in Equation 1 and is approximately equal to 3.5 trillion.

$$\sum_{n=4}^{7} 62^n \approx 3.5 \text{ trillion} \qquad (1)$$

Tinyurl.com has a shortest hash length of 3 and a longest hash length of 7. Every observed URL contained only lowercase alphanumeric characters making for a total of 36 characters. Hashes can therefore also be expressed as a base-36 integer. The total key space is shown in Equation 2. `TinyURL` also offers the possibility of creating your own alias as described in Section 5.3. This functionality will not be part of the research.

$$\sum_{n=3}^{7} 36^n \approx 80 \text{ billion} \qquad (2)$$

Goo.gl shortens URLs to hashes of 4-6 digits in length [22]. The hash consists of all alphanumeric characters, both upper- and lowercase. Hashes are numbers in the same base as `bitly`, base-62. The key space is shown in Equation 3 and is approximately equal to 56 billion.

$$\sum_{n=4}^{6} 62^n \approx 58 \text{ billion} \qquad (3)$$

5.3 Features

The three chosen services differ in what they offer in terms of API, scanning of short URLs for malicious content and rate limit. This section explores the differences encountered between services and how they were dealt with during the research.

TinyURL is by far the most simple service of the chosen three. It is a one page site where users can enter a URL and shorten it in one of two methods. First its users can let `TinyURL` generate the short hash from a set as described in Section 5.2 or they can create an alias such as:

`http://www.myisp.com/~MyUsername/index.html`

[7]Ranked as the 28th most visited site globally as of 26th of January 2016

[8]Accessed 5th of January 2016

to:

`http://tinyurl.com/MyUsername`

This functionality makes it possible for users to personalize their URL.

`TinyURL` does not publish an API, rate limit or policy regarding the scanning of URLs for abuse. It only briefly touches the subject of malicious links in their Terms of Service; stating that they will report any misuse of their service to ISPs and or governments[23]. As a safety measure for visitors they make it possible to preview a `TinyURL` link by prepending `preview` in front of the link like so: `http://preview.tinyurl.com/XYZ123`. The service does not provide statistics about clicks, referrals or lifetime of the short URLs.

In practice this study encountered no rate limit whilst attempting to brute force the short hashes of `TinyURL`.

Bit.ly used to be the go to service for all Twitter short URLs. Founded in 2008 it states on its site that it has shortened around 24.5 billion URLs[9]. It provides users with more services compared to `TinyURL`. It has published a number of topics on their knowledge base and website about API functionality, abuse policy and rate limits.

The most important feature for our research is the fact that `bitly` actively works together with Google, Facebook and Twitter to counter abuse [4]. Stating that they actively test URLs against the Google Safe Browsing service and share information with Facebook and Twitter. We confirmed this by inputting a link which we knew was malicious. When attempting to share such a `bitly` link, the user will be redirected to a landing page stating that the site contains malware.

Like `TinyURL`, `bitly` offers a preview feature. By appending "`+`" to the URL, users can view the site without really visiting it.

The published rate limit states that users make up to 1 million queries per API key per day. This means that potentially 1 million links could be stored in the database per day, however the key space of `bitly` is by far the largest. As stated in Section 5.2 the total key space is approximately 3.5 trillion URLs.

`bitly`'s API supports various types of functions that make data available about each link. This data is publicly available and contains information such as; referrers, clicks and various other metrics [3].

Goo.gl was launched towards the end of 2009 as a shortening service for Google's own links. From september 2010 onwards it became available to the wider user space [27]. The `goo.gl` service provides users with similar functionality as `bitly` including a preview method like `bitly`. The published API limit is 1 million queries per day [8], however during the research we found out that the API rate limit was more stringent. It limits requests to 1 per IP/second/user API key. Theoretically this means that 86400 requests are possible per day, however this was not achieved. Only 0.8 requests per second were possible, making it hard to scan the `goo.gl` hash space effectively.

As well as the limit on the API the `goo.gl` service has a limit on the amount of HTTP requests on its redirection web servers. This is along the lines of the API rate limit but used solely the IP address as the identifier. If queries go faster than approximately 1 request per second, per IP address, the crawler received a `403 (forbidden)` response citing rate limits. The API supports a simplified amount of functions compared to the `bitly` API, however the information of interest such as clicks and referrers is available.

`goo.gl` returns a "status" field in their response indicating the malware classification of the long URL. Although not explicitly stated as being the same as Google Safe Browsing, the "status" values are very similar to the Google Safe Browsing API responses; Phishing, Malware and Unwanted [8].

Similarites between services
One thing that the three services have in common is that they do not destroy or remove any of the old URLs. This means that a short URL that was created a long time ago will never be marked inactive or deleted even though the long URL it expands to is no long active or online. All services use permanent redirects to redirect a visitor to the long URL and a temporary redirect to a landing page. In addition to this, when a browser does not follow the HTTP redirects all services present the visitor with a page with a hyperlink to the long URL.

6 Methodology

The following subsections describe how the data collection took place, what data was collected, what information was left out to meet privacy concerns and how the collected data was analyzed.

6.1 Data collection

A simple script was developed to generate short URL hashes. It takes a random number, generated using NumPy[10] and then converts it to the appropriate base of alphanumerical characters. Assuming the short URL service providers distribute the long URLs evenly over the key space, the script

[9]As of January 2016

[10]`http://www.numpy.org/`

creates an evenly distributed random sample over the whole key space. It is known that the short URL providers do not utilise the whole key space. Therefore, the generated short hashes will not all point to a long URL.

The generated short hashes are read by a second program which collects data about both the long- and short URL and stores it in a database. For the short URL, if the data is available from the service provider, metadata like the Creation Time, the number of times the short URL was clicked, which user created the short URL and which sites refer most to this URL is collected. In Figure 1 a flowchart of the data collection process is shown and Section 7 describes the developed software as well as the hardware setup in more detail.

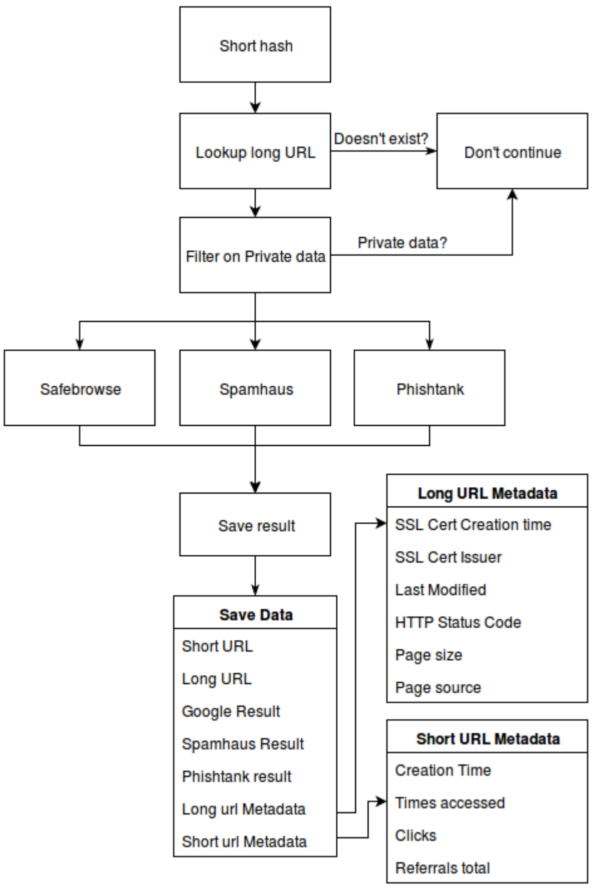


Figure 1: Program structure

Private Data Filter This research characterizes private data as; the protocol of the URL, login and password information or sensitive query strings in the URL path. If the protocol was not HTTP or HTTPS the URL was discarded. The actual domain name could not start with a `<user>:<password>@` construction and if parameters which were akin to user and/or password existed in the URL path, the short URL was discarded as well.

Metadata collection When a URL has passed the Filter stage the program tests the long URL against the Google Safe Browse API, the SpamHaus API and the PhishTank database. Metadata is collected about both the long and short URLs if they achieve a hit. Table 1 shows what metadata the program collects. This information makes it possible to analyze traits and correlation between malicious short URLs.

Short URL	**Long URL**
Creation Time	Last Modified header
Times accessed	HTTP Status Code
User creation ID	SSL Certificate
Referrals	Site Size

Table 1: Metadata stored for both URL types

6.2 Testing against blacklists

While gathering data the program uses 3 different services to determine whether a URL is malicious. A URL is flagged when one of the services gives a positive hit. Below the chosen are explained.

Google Safe Browsing is a blacklist maintained by Google. It allows a developer to download the whole database and perform incremental upgrades to that database, thereby allowing a virtually unlimited amount of API calls. There are three lists. The malware list (`goog-malware-shavar`), the phishing list (`googpub-phish-shavar`) and a general unwanted list (`goog-unwanted-shavar`). Once a domain is flagged it remains in the Safe Browse database until it has been confirmed to be clean or if there have been no new reports about the domain for 90 days.

SpamHaus originates as a blocklist for IP spaces which send out spam. Later on, they added a domain blocklist. By performing a DNS Lookup for the IP address at `<ip>.zen.spamhaus.org` and the domain name at `<domain>.dbl.spamhaus.org` it is possible to check whether a long URL is in the SpamHaus blocklist.

PhishTank was used by Maggi et al[7] in order to classify long URLs. Moreover, in [20] the authors measured a 97% accuracy in PhishTank's database. While conducting this study the PhishTank database contained about $28,500$ entries. New entries were automatically added every 4 hours.

6.3 Analyzing the data

By storing the data in a standardised way we can analyze the statistical differences between groups.

Using a two-sample unpaired t-test it is possible to compare means and proportions between two samples. It appeared during the study that one site was abundantly present in our sample of malicious URLs, therefore we decided to investigate further into this site by analysing the redirect chain.

7 Experimentation

This section describes the setup of the conducted experiments. First the Data Gathering phase is described, then the setup for the taxonomy of `asterpix.com`.

7.1 Data gathering

A Python script[11] was created to gather the data as described in Section 6.1. To run it needs a directory of files with one short hash per line. We generated this with our own URLGenerator[12]. The program was then concurrently ran with 8 threads on 12 Virtual Machines (VM). The total amount of concurrent threads we ran was 96. The VMs were spread evenly over two rack servers. The servers both had 8GB of RAM memory and an Intel Xeon L3426 CPU clocked at 1.86GHz. On one of the servers a MongoDB v2.6.3 instance was ran for the data collection and PhishTank testing progress. During experimentation the PhishTank database was updated every 4 hours. The Google Safe Browsing database was updated every 2 hours. Using this setup, data was collected starting from January the 15th through January the 18th 2016. All gathered data is saved in a uniform format. We chose to define a data object in JSON. An example of such a document is found in Appendix C.

7.2 Asterpix.com

In this research the `asterpix.com` website was requested with a virtual machine running on VirtualBox[13] 5.0.14. The VM ran `Ubuntu 14.04 desktop` and was connected to the Internet via a number of Virtual Private Networks (VPN). We used `airvpn.org` endpoints in Asia, America and Europe to visit the site and observe to where the user would be taken. Table 2 shows the endpoints that we used during the observations.

Within the virtual machine, `asterpix.com` was retrieved a number of times with a private browser window. When no new redirects would present themselves or if the malware app always redirected to the same site the investigation ceased.

The browser used was Firefox with the following user agent string:

AirVPN	**IP address**	**Geo Location**
Europe	213.152.161.133	the Netherlands
Asia	103.254.153.99	Singapore
America	104.254.90.187	Canada

Table 2: IP addresses of the AirVPN endpoints and Geo location according to `geoiptool.com`

```
Mozilla/5.0(X11;Ubuntu;Linuxx86_64;rv:
40.0)Gecko/20100101Firefox/40.0
```

To trace the redirect chain we use `Wireshark 1.12.1` to follow DNS requests and HTTP streams. With this method content of packets are dumped and examined to follow all referrals and JavaScript redirects.

8 Results

In this section we will present the observed results. In Section 8.1 we will present numbers about the gathered data and in Section 8.2 we will present our observations about the malicious entrypoint `asterpix.com`.

8.1 Dataset statistics

This section will describe the statistics that can be drawn from the numbers that were observed during the research. The terms used to classify long URLs in the following sections are defined in Section 3.

8.1.1 Absolute numbers

In total we gathered information about $1,388,006$ short URLs. The most URLs were collected from `TinyURL`: $1,377,969$. From `bitly` we received $5,915$ URLs and from `goo.gl` the crawler gathered $4,122$ URLs. The main reason for this difference was due to the various rate limits on the services as described in Section 5.3.

When taking into account the definitions of "detected" and "undetected" as described in Section 3, of all URLs gathered, a total of 948 were flagged undetected. The majority of undetected URLs, 946 came from `TinyURL`. The rest, 2, came from `bitly`. Both `bitly` URLs interestingly also point to `asterpix.com`. At `goo.gl` there were no undetected hits. `TinyURL` will redirect visitors to a landing page when a URL is detected and marked malicious. From our dataset it appeared that `TinyURL` had detected $70,302$ links in the past. `bitly` has 1 detected link and `goo.gl` has 4. The numbers are shown in a more convenient way in Table 3.

In Table 4 a list of hostnames from sites which are flagged as malware and the amount of times they appear in the database is shown. It is clear

[11] `https://github.com/OS3RP1Crawler/crawler`
[12] `https://github.com/OS3RP1Crawler/hash_generator`
[13] `https://www.virtualbox.org/`

Service	URLs	Undetected	Detected
TinyURL	1,377,969	946	70,302
Bit.ly	5,915	2	1
Goo.gl	4,122	0	4
Total	**1,388,006**	**948**	**70,307**

Table 3: Numbers in our dataset, column headers as defined in Section 3

.

that "asterpix.com" is a large contributor to our malware dataset. We encountered `asterpix.com` in both of the services that had malware hits, `TinyURL` and `bitly`. Further research into Asterpix is described in Section 8.2 The last entry is a combined entry of all hostnames which occur less than 4 times in the database. The full table is shown in Appendix A.

Hostname	Count
www.asterpix.com	495
video.asterpix.com	113
www.tagvn.com	75
www.filelodge.com	57
keyknowhow.com	23
hurl.content.loudeye.com	16
static.zangocash.com	14
www.perfectporridge.com	13
www.content.loudeye.com	5
Small counts (<= 4)	137

Table 4: Count of hostnames of malicious sites appearing in our sample

For testing we used a number of different hash lengths to put in the path of the short URL. In Figure 2 a comparison is made between the different lengths. Shorter hashes often indicate that the short URL was generated further in the past. The lengths depicted in the figure refer to active short URLs, not hashes which have not (yet) been assigned. New `TinyURL` URLs generate with a hash length of 6, new `bitly` URLs have a hash length of 7 and new `goo.gl` URLs have a hash length of 6.

8.1.2 Malware statistics

In our dataset PhishTank did not flag any links as malicious or spam. Only one domain in our sample was found in the SpamHaus Domain blacklist. This domain was `tinyurl.com`. Every hit from the Google SafeBrowsing API was for the "`goog-malware-shavar`" list.

None of the flagged links were SSL/TLS links, thus we did not gather any certificate (meta-)data of malicious sites.

TinyURL malware

As mentioned before, `TinyURL` does malware filtering itself as well. Links which have been flagged as malware by `TinyURL` redirect to a landing page with the URL

`http://tinyurl.com/nospam.php?id=<hash>`

From this page it is impossible to retrieve the original long URL. On the "`nospam`" page it says that the user used the URL in violation of the terms of service.

Summing our malware hits and `TinyURL`'s malware hits we get a total of $946 + 70,302 = 71,248$ malware hits. Taking the total amount of crawled `TinyURL` URLs we get a percentage of:

$$\frac{71,248}{1,377,969} * 100\% = 5.17\%$$

malware. If the sample is uniformly distributed we can say that the total percentage of malware on `TinyURL` is equal to 5.17 ± 0.04 with a confidence level of 95%.

Length of short hashes collected

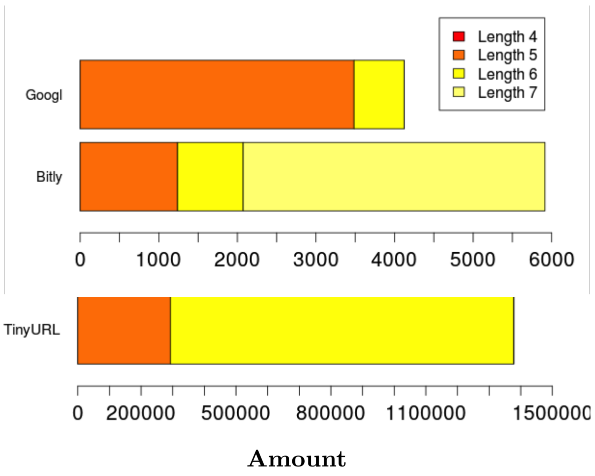


Figure 2: Hash lengths that point to long URLs per service

Malware proportion

`Bit.ly` has a total of 3 detected and undetected malware links. The total percentage of malware on `bit.ly` is

$$\frac{3}{5915} \approx 0.05$$

So the percentage of malware in the whole keyspace of `bit.ly` lies within $0.05\% \pm 0.06\%$. For `goo.gl` we observed a total of 4 malware hits. Thus, the percentage of malware on `goo.gl` is

$$\frac{4}{4,122} * 100\% \approx 0.10\%$$

From this we can say with a 95% confindence interval the value of the population lies within $0.10\% \pm 0.1\%$.

Undetected malware proportions

In both bitly and TinyURL one or more URLs were undetected. We hypothesise that the proportion of undected malware in TinyURL $= \mu_0$ and bitly $= \mu_1$ is equal.

$$\text{Null hypothesis: } \mu_0 = \mu_1$$
$$\text{Alternative hypothesis: } \mu_0 \neq \mu_1$$

The exact proportions for undetected malware are for bitly:

$$\mu_0 = \frac{2}{5915} \approx 0.00034$$

and for TinyURL:

$$\mu_1 = \frac{946}{1377969} \approx 0.00069$$

Using Welch's t-test for unpaired, two-sample t-tests we get a p-value of 0.1468. This is well above the chosen significance level $\alpha = 0.05$, thus we do not reject the null hypothesis and say that both TinyURL and bitly have proportionally equal amounts of undetected URLs. Goo.gl did not have any undetected URLs in our sample.

The next statistic is about the total number of malicious sites in the sample. TinyURL has a lot more malicious URLs than bitly. Therefore, we hypothesise that the difference in means is statistically significant.

$$\text{Null hypothesis: } \mu_0 = \mu_1$$
$$\text{Alternative hypothesis: } \mu_0 \geq \mu_1$$

From our sample we collect that

$$\mu_0 = \frac{71248}{1377969} * 100\% \approx 5.17\%$$

and

$$\mu_1 = \frac{3}{5915} * 100\% \approx 0.05\%$$

Using Welch's t-test for an unpaired, two-sample t-test we get the p-value $< 2.2e - 16$. This is much smaller than the significance level $\alpha = 0.05$, so we reject the null hypotheses and state that TinyURL points more often to malicious URLs than bitly does.

For goo.gl our data sample shows that

$$\mu_2 = \frac{4}{4122} * 100\% \approx 0.10\%$$

Performing the same test for goo.gl $= \mu_2$ and bitly $= \mu_1$ we get a p-value of 0.4136, which makes us accept the null hypothesis, indicating that proportionally the amount of malware on goo.gl and bitly is the same for a confidence level of $\alpha = 0.05$.

8.2 Asterpix.com

As discussed in Section 8.1.1 we encountered a large number of hits on the site asterpix.com. This site, first registered on "October 05, 2006", according to who.is[14] has had a number of forms in its history. This section briefly reviews the history of the site, investigates why this site has such a large number of hits in the dataset and observes how locality of the visitor influences the nature of how they are redirected from asterpix.com towards the various the endpoints of the redirect chain.

8.2.1 History of asterpix.com

asterpix.com used to be a video sharing site. The internet archive[15] shows upon inspection that from its creation in 2006 the site has had multiple forms.

2006-2012

In the period from 2006 to 2012 asterpix.com was a site that offered a number of services. As demonstrated in Figure 3 its first appearance was as a video sharing site that enabled users to import their videos and create links to external sources framed on top of the video.

From 2009 onward the site shifted towards a local market place and search tool. It remained that way until the beginning of 2012. From that moment on the Internet archive records that the site goes blank.

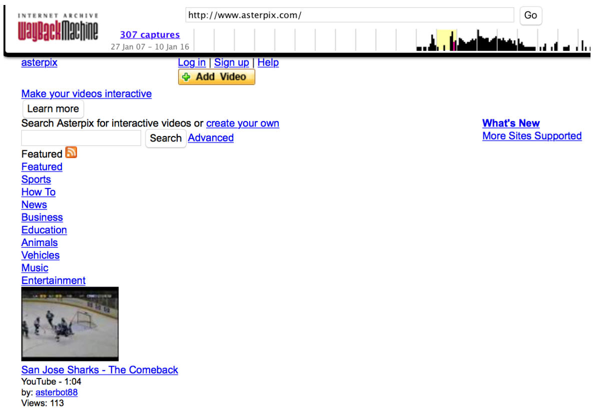

INTERNET ARCHIVE WayBackMachine
http://www.asterpix.com/ Go
307 captures
27 Jan 07 – 10 Jan 16
asterpix
Log in | Sign up | Help
Add Video
Make your videos interactive
Learn more
Search Asterpix for interactive videos or create your own
Search Advanced
What's New
More Sites Supported
Featured
Featured
Sports
How To
News
Business
Education
Animals
Vehicles
Music
Entertainment
San Jose Sharks - The Comeback
YouTube - 1:04
by: asterbot88
Views: 113

Figure 3: Asterpix screenshot from the internet archive from nov 2008[14]

2012-2016

From 2012 to mid 2015 the Internet archive shows that asterpix.com becomes a redirect site. During this period the sites more dubious nature evolves. It gives the user temporary redirects to various sites that are simply landing pages for registrars or domain squatting.

Mid 2015 asterpix.com appears in the Google Safe Browsing list as a malware site[13]. It is

[14]https://who.is/whois/asterpix.com
[15]http://web.archive.org

hosted in Australia in a part of the Trellian network[16], according to who.is. The IP to which DNS resolves, hosts a number of sites that have a bad reputation[13] such as www.malboro.com[17] and wwwmyciti.com[18]. Cymon.io security threats framework proves that the IP is part of a subnet of the Trellian network that is hosting abuse.

8.2.2 Large number of hits

In Table 4 we observe that the largest amount of links that were flagged during the data gathering phase are from either www.asterpix.com or video.asterpix.com. For the most part we cannot exactly determine when the urls are created. However the two malware hits from the bitly service, as stated in Section 8.1.1, also expanded to asterpix.com URLs. These links were created in August 2009 in the period that asterpix.com manifested itself as a video service.

In total bitly registered one click on these two links. It is not known due to the lack of an API how many clicks the TinyURL links registered. The lack of clicks indicates that the bitly links are not actively abused.

The expanded short URLs that the crawler visited have the following structure:

```
http://www.asterpix.com/console/?avi=<ID>
```

or

```
http://video.asterpix.com/v/<ID>/<Title>/
```

Whilst browsing the Internet Archive we encountered that all videos that were added to the database of asterpix.com received a URL such as described above. This would mean that the short URLs that were crawled are older than 5 years, but still active even though the function of the site changed multiple times during the course of its history.

It seems that during its past a significant amount of asterpix.com URLs have been shortened. Even though the function of the site has changed and the long URLs are dead, users are still exposed to the dangers of this site, as they are not protected by a landing page of the short URL provider.

8.2.3 On the nature of asterpix.com

Upon visiting any URL of asterpix.com the visitor enters a sophisticated redirect chain whose nature is very similar to the case study of blackraybansunglasses.com in Lee et al[16]. However, it depends greatly from where in the world a user visits the site.

[16]http://www.trellian.com
[17]Should be: http://marlboro.com
[18]Should be: http://www.myciti.com

We assume that asterpix.com is the entry point to a malware network as this is the domain that is flagged by Google Safe Browse. However it could easily be one of a multitude of entry points. By taking a step back it is possible to describe the global flow of the redirection network as follows: Entry/Evaluation, Redirection, Hand off. Below we analyze the redirect chains according to the three categories

Entry/Evaluation

Upon entry to the network we observed that the visitor will either enter the network or be deferred to a non malicious landing page. This decision is taken at asterpix.com. Depending on how the visitors are evaluated at the entry point, they will follow along the redirection chain to one of eight encountered hand off points.

First asterpix.com evaluates visitors on the basis of their location. The second step in the evaluation of the visitor is to check if they have visited the site before. We observed during experiments that it depended greatly on how often the user visited the site, to what they would be redirected to. If asterpix.com is visited a few times in short succession visitors are taken to a non malicious page. In the case of a first visit, the entry point to the adware network evaluates the geolocation of the visiting IP address. Based on this information it directs the visitor to the next location.

As well as country and language, asterpix.com also makes a decision about which hand off site should be visited within that geolocation. Different geolocations offered a number of different sites to visitors. During experiment a number of redirect paths observed that take the vistor to the eventual endpoint. The endpoints are distinct per location however the chain towards the endpoint show some overlap. This research identified three different second stage sites to which visitor could be referred to and two different non malicious landing pages.

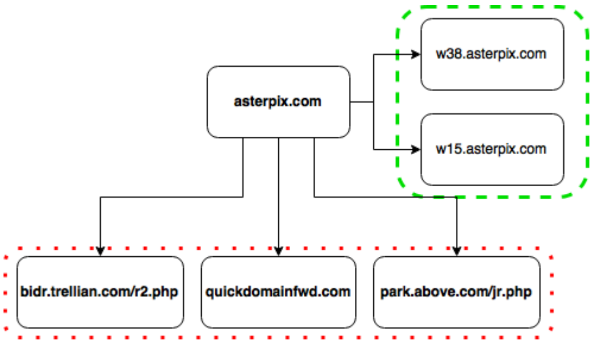


Figure 4: This Figure shows how a user will be redirected after evaluation. The sites highlighted in the dashed box are non malicious landing pages. The dotted box represents the start of malicious redirect chains

In Figure 4 we see two landing pages;

ww15.asterpix.com and ww38.asterpix.com. If the the site detects that users have visited the site multiple times within a short period and if users come from an Asian or European node, the application directs users to a non malicious page that displays generic information. Asian users see ww15.asterpix.com and European users see ww38.asterpix.com. During experiments the American endpoint always got directed into the chain.

The three domains that are highlighted within the dotted box in Figure 4 represent the three malicious entrypoints in the redirect chain. Whilst analysing Domain Name System (DNS) information we see that bidr.trellian.com and park.above.com both resolve to Trellian Pty located in Australia and quickdomainfwd.com resolves to the Confluence Networks who are registered in the British Virgin Isles.

When looking at the commonalities between these sites we see that quickdomainfwd.com is registered in Queensland, Australia. It is therefore likely that the hub of this network is also located in Australia.

Redirection

The previous paragraph discussed the strategy that asterpix.com uses to evaluate visitors to lead them down the chain of redirection. This paragraph will describe the characteristics of the chain within the asterpix.com network.

The principal method to route visitors with JavaScript within the redirect chains is, the "Unescape" and "String Manipulation and Eval" as described by Chellapilla et al [6]. In the majority of observed cases the visitor will be passed to either park.above.com or bidr.trellian.com. These two sites then pass their visitors along a chain of three or four hops to the hand off target. The principle secondary re directors are zeroredirect1.com and zeroredirect2.com. They served the American and European hub, passing the visitor along to different hand off sites. The strategy they used to identify hand off targets was as follows:

```
z[a-z].zeroredirect1.com/zcredirect
```

to

```
z[a-z].zeroredirect2.com/zcredirect
```

The second letter of the sub-domain identifies the route the visitor will take, along with the parameters that are passed in the path of the URL. For a full depiction of the observed redirect chains see Appendix B.

None of the secondary or tertiary steps in the redirect chains are flagged by the lists stated in Section 6.2 during the period of experimentation. This resulted in an uninterrupted chain from asterpix.com to the hand off site.

Hand off

Analysis of the hand off sites show that they contain the usual scam and ad ware that can be expected. A number of sites advertised the opportunity to win large sums of money and two other sites tried to convince users that by filling in a survey they could win an iPhone 6s. One site advertised the possibility to purchase products such as TVs, tablets and computers for a few dollars. Table 5 shows a rudimentary categorization of the encountered hand off sites.

If the visitor successfully arrived at a hand off site, they will encounter a site that is catered to their locality. The hand off sites from asterpix.com attempt to present their sites in the language of the visitor and represent companies in the visitors country. For example: One of the European hand off sites presents the visitor with a voucher for 500 € at the Netherlands' largest grocery store in the Dutch language. Secondly: One of the Canadian sites targeted visitors by impersonating a large Canadian company. Thirdly: the Asian endpoint led its visitors to a Singapore specific Facebook page.

9 Conclusion and Discussion

This section draws conclusions from the gathered data and observations made during the experiments. After that it discusses the results, by exploring some caveats encountered during the research.

9.1 Conclusion

In Section 2 this paper proposes two sub questions. With the data collected during experimentation it is possible to formulate an answer to these questions.

Statistical analysis

The first question asked: which service provides proportionally the most malicious short URLs. Section 8.1.2 shows through statistical analysis, that both bitly and TinyURL have a insignificant difference, in the proportion of undetected malicious sites. goo.gl had no undetected malware hits at all. We also observed from the samples that TinyURL has a significant greater portion of malicious links than bitly when both populations are compared. The difference between malware hits (both detected and undetected) in bitly and goo.gl is not statistically significant. Due to the nature of the found malware links in TinyURL and their correlation to the bitly links, we can see the

Site	Categorisation
rewardsurveybrands.com	Online survey/selling of high end products
facebook.com/campaign	Unknown
tracking.broker-science.com/lp-millionaires...	Win large amounts of cash
cash-o-matic.com	Win/borrow large amounts of cash
supermarket-vouchers.com	Win vouchers for supermarkets
bhQzz.exclusiverewards.qodo.info	Online survey with reward

Table 5: Malware sites and type

links originate in 2009, when Asterpix was a legitimate site. This implies that both `TinyURL` and `bitly` do not scan their old keyspace for long URLs which became malicious over time.

Scanning of old URLs

As mentioned in Section 5.3 none of the providers recycles its short hashes. Even though they actively flag their own URLs upon creation, in the case of `TinyURL` the results indicate that `TinyURL` may not to check old active URLs. The `asterpix.com` case shows that links that are "safe" upon creation do not necessarily stay safe later. Eventhough `asterpix.com` has been flagged as malicious for over 10 months[13], `TinyURL` has not flagged them as malicious.

Longevity of short URLs

As stated in Section 4.2, previous research has indicated [2] that after two years more than 99% of all short URLs are inactive. The case of `asterpix.com` however, may be an unforseen risk in the manifestation of short URLs. When considering the data gathered by this research about `asterpix.com` and considering that it is highly likely that asterpix's short URLs are older than five years. We see that this case is one of the few short URLs that remain online after two years, but at the same time they are a significant proportion of the URLs that were flagged by the crawler. This raises the following question: What percentage of URLs that are older than two years, are still active and contain malware? As short URL providers do not remove broken or dead short URLs if the site they point to goes off line or changes function, this fundamentally changes how the Internet and Hyperlinking works as also stated in Antonaides et al[2]. As the short URLs show signs of link rot, cases like `asterpix.com` could increase in the future as these domains are re purposed.

Blacklisting redirect sites

One thing noted in Section 8.2.3 is that none of steps in the redirect chain after `asterpix.com` are flagged as malicious. We propose that it would be a significant degradation of the effectiveness of malware networks, if underlying redirect chains are also flagged in services like Google Safe Browsing. As Google Safe Browse functionality is embedded in modern day browsers, blocking redirect domains could be even more effective as the user will be prompted to accept the malware warning numerous times. Another benefit in blocking the underlying re-directors is that if they are also used in other, undetected malware network entry point sites like `asterpix.com`, these entry points may be rendered less effective.

To the extent of our knowledge this research is the first documentation of a malware network that uses geolocation to target visitors. By targeting visitors in their native language and by masquerading as companies local to the user, this campaign is designed to yield more results for the proprietors of the network. Analysis of the endpoint sites, tells us that they are very specific to the location of the the IP address of the user. It could indicate that we have only scratched the surface of total size of the `asterpix.com` network. However, the case of `asterpix.com` shows that multiple geolocations use the same underlying redirect domains. If we assume that the network of `asterpix.com` shares redirect infrastructure with other malicious entry points, this further strengthens the case to block underlying redirect sites, as it will greater impact in protecting users of the Internet.

Answering the main question

Recall that our research question is "What portion of the short URL services are used for malicious purposes and what does the abuse look like?" Using answers from the previous sections we formulate an answer to the main research question.

In `TinyURL` we observed a rather large portion of 5.17% previously or currently malware related links in our sample. The portions in `bitly` and `goo.gl` were 0.05% and 0.01% respectively. This suggests that the `TinyURL` service is used significantly more to obfuscate malicious long URLs.

For the second part of this question, our sample shows that large portion of the encountered malicious URLs, are for a site called `asterpix.com`. By investigating `asterpix.com` we observed a sophisticated adware network which offers content to you based on your geolocation.

9.2 Discussing the results and limitations of the research

In earlier work Gupta et al [12] stated `bitly` does not remove malicious URLs, but only shows a warning page. However, contrary to `TinyURL` the warning page at `bitly` can be passed. This allows users to click through and visit the flagged site anyway. Egelmann et al [9] observed that 79% of all visitors choose to heed active warnings. To protect the remaining 21% of visitors who do not heed warnings, we propose that `bitly` adopts a similar approach to malicious links as `TinyURL`. This assumes that the URL classification services have a low false positive rate as stated in Whittaker et al [25] , as to not create detrimental effects on the usability of the URL shortening service.

Technical Limitations

Google has a very strict rate limiting policy. It allows just 1 request per second per user. During crawling it became apparent that we were hitting rate limits at both Google and `bitly`. The sample sizes of both `goo.gl` and `bitly` are therefore not as large as `TinyURL`'s. One could improve on this by using more API keys and more Virtual Machines with more IP addresses.

As well as rate limiting the theoretical search space of `bitly` is extremely large with around 3.5 trillion possible hashes, as stated in Section 5.2. This means that statistical analysis of data could be difficult as we need assume that `bitly` evenly uses their key space. We do not exactly know how the service providers distribute the keys. Do all short URLs that are created at this time have a length of 7 for the hash, or 5, or 6?

An important detail in the rate limits that Google and `bitly` employ is that they are on all queries and HTTP requests. This research indiscriminately brute forced the theoretical key space of the providers. Especially with `goo.gl` these limits are stringent. As stated above more knowledge about the build up of the key space and how it is employed will help further research.

TinyURL allows users to preview a shortened URL by default, but our program did not cater for this result; it did not investigate what the preview URLs expanded to. The amount of preview URLs is $21,702$ in our dataset and they may or may not be malicious.

Most data came from `TinyURL`. This means that statistical analysis of the data from the `TinyURL` service is inherently stronger compared to the other service providers.

Blacklisting redirect sites

As stated in the conclusion of this research blacklisting redirect domains may be an effective way of blocking malware networks. In stating this we are assuming that by adding this extra layer of protection visitors will decide to heed the secondary warning that they will receive. Further research is needed into the effectiveness and feasibility of such a method.

Locality

Unfortunately it is likely that this research was only able to visit a small number of the localized sites that are active within the malware network behind `asterpix.com`. Next to that we do not know if there are any other entry point sites that share characteristics or influence the targeting of visitors in any way.

Longevity of URLs

It is unknown if the longevity of short URLs is a risk for Internet users. This research observed how relatively old short URLs may be re-purposed for malicious purposes. It is however was not possible to see if in the case that was researched it is actively abused. We do no know if the malicious short URLs receive any clicks, because `TinyURL` does not provide an API.

9.3 Summary

To conclude we believe that this research contributes to the observation of abuse in short URL services. It has seen that service providers are actively checking input of users to increase the safety and frustrate the activity of abusers. However it may be that older URLs that become malicious after a period of being dormant, stay under their radar. `Asterpix.com` being our case in point.

Next to these facts we also uncovered an adware network behind a malicious site that serves as an entry point to a sophisticated network. This network behaves in a way that leads us to believe it targets users on the basis of their geolocation. This finding is not surprising considering the global usage of the Internet. It is however a topic which is interesting to explore as uncovering weaknesses in networks such as the one this study observed, may help to increase the effectiveness of blacklisting services.

10 Future Work

In this section we identify a number of subjects for future work:

- How often do short URLs links go offline and get repurposed like the case of `asterpix.com` and what proportion are malicious?
- Test the feasibility of blocking secondary and/or tertiary redirect domains and examine

the effects on both the visitors and malicious network.

- A further case study into locality based malware networks by looking at; entry points, end points and what redirect chains they have in common.
- Optimization of the search of the keyspace of `bitly` and `goo.gl`.
- Look into smaller and lesser known providers and compare their filtering mechanisms to the larger providers.

Acknowledgements
The authors are thank full for the support of the proofreaders and reviewers of this paper, in particular Ruud Verbij at KPMG. This work has also been supported by Tom Curran who provided the VPN services used during the research.

References

[1] Sebastian Anthony. *In 2015 tablet sales will finally surpass PCs, fulfilling Steve Jobs post-PC prophecy.* 2014. URL: `http://www.extremetech.com/computing/185937-in-2015-tablet-sales-will-finally-surpass-pcs-fulfilling-steve-jobs-post-pc-prophecy` (visited on 01/08/2016).

[2] Demetris Antoniades et al. "We.B: The Web of Short Urls". In: *Proceedings of the 20th International Conference on World Wide Web.* WWW '11. New York, NY, USA: ACM, 2011, pp. 715–724. ISBN: 978-1-4503-0632-4. DOI: `10.1145/1963405.1963505`.

[3] *API Documentation.* URL: `http://dev.bitly.com/api.html` (visited on 01/08/2016).

[4] Bitly knowledge base. *How can I be sure a Bitly link is safe to click on?* 2016. URL: `http://support.bitly.com/customer/portal/articles/1765839?b_id=5612&t=501703` (visited on 01/08/2016).

[5] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax.* RFC 3986 (INTERNET STANDARD). Updated by RFCs 6874, 7320. Internet Engineering Task Force, Jan. 2005. URL: `http://www.ietf.org/rfc/rfc3986.txt`.

[6] Kumar Chellapilla and Alexey Maykov. "A Taxonomy of JavaScript Redirection Spam". In: *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web.* AIRWeb '07. New York, NY, USA: ACM, 2007, pp. 81–88. ISBN: 978-1-59593-732-2. DOI: `10.1145/1244408.1244423`.

[7] Sidharth Chhabra et al. "Phi.Sh/$oCiaL: The Phishing Landscape Through Short URLs". In: *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference.* CEAS '11. New York, NY, USA: ACM, 2011, pp. 92–101. ISBN: 978-1-4503-0788-8. DOI: `10.1145/2030376.2030387`.

[8] Google Developers. *URL Shortener.* URL: `https://developers.google.com/url-shortener/v1/getting_started` (visited on 01/08/2016).

[9] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. "You'Ve Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '08. Florence, Italy: ACM, 2008, pp. 1065–1074. ISBN: 978-1-60558-011-1. DOI: `10.1145/1357054.1357219`.

[10] R Fielding et al. *Hypertext Transfer ProtocolHTTP/1.1, RFC2616, June 1999.* RFC 2616. IETF, June 1999. URL: `https://tools.ietf.org/html/rfc2616`.

[11] Sujata Garera et al. "A Framework for Detection and Measurement of Phishing Attacks". In: *Proceedings of the 2007 ACM Workshop on Recurring Malcode.* WORM '07. New York, NY, USA: ACM, 2007, pp. 1–8. ISBN: 978-1-59593-886-2. DOI: `10.1145/1314389.1314391`.

[12] N. Gupta, A. Aggarwal, and P. Kumaraguru. "bit.ly/malicious: Deep dive into short URL based e-crime detection". In: *Electronic Crime Research (eCrime), 2014 APWG Symposium on.* Sept. 2014, pp. 14–24. DOI: `10.1109/ECRIME.2014.6963161`.

[13] eSentire Inc. *Cymon.io.* URL: `https://cymon.io/domain/asterpix.com` (visited on 01/22/2016).

[14] Wayback Machine Internet Archive. *The Internet Archive.* URL: `http://web.archive.org/web/20081108030257/http://www.asterpix.com/` (visited on 01/22/2016).

[15] Florian Klien and Markus Strohmaier. “Short Links Under Attack: Geographical Analysis of Spam in a URL Shortener Network”. In: *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*. HT ’12. New York, NY, USA: ACM, 2012, pp. 83–88. ISBN: 978-1-4503-1335-3. DOI: `10.1145/2309996.2310010`.

[16] Sangho Lee and Jong Kim. “WarningBird: Detecting Suspicious URLs in Twitter Stream.” In: *NDSS*. 2012.

[17] John Levine. *DNS blacklists and whitelists*. RFC 5782. IETF, Feb. 2010. URL: `https://tools.ietf.org/html/rfc5782`.

[18] Federico Maggi et al. “Two Years of Short URLs Internet Measurement: Security Threats and Countermeasures”. In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW ’13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 861–872. ISBN: 978-1-4503-2035-1.

[19] Niels Provos Panayiotis Mavrommatis and Moheeb Abu Rajab Fabian Monrose. “All your iframes point to us”. In: *USENIX Security Symposium*. 2008, pp. 1–16.

[20] Tyler Moore and Richard Clayton. “Evaluating the Wisdom of Crowds in Assessing Phishing Websites”. English. In: *Financial Cryptography and Data Security*. Ed. by Gene Tsudik. Vol. 5143. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 16–30. ISBN: 978-3-540-85229-2. DOI: `10.1007/978-3-540-85230-8_2`.

[21] Sushant Sinha, Michael Bailey, and Farnam Jahanian. “Shades of Grey: On the effectiveness of reputation-based blacklists”. In: *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*. IEEE. 2008, pp. 57–64.

[22] Alexandros Stavroulakis, Xavier Torrent Gorjon, and Nikolaos Triantafyllidis. “Exhaustive Search on URL Shorteners”. unpublished. 2014.

[23] tinyurl.com. *Terms of use*. URL: `http://tinyurl.com/#terms` (visited on 01/08/2016).

[24] Twitter. *t.co links*. 2016. URL: `https://dev.twitter.com/overview/t.co` (visited on 01/08/2016).

[25] Colin Whittaker, Brian Ryner, and Marria Nazif. “Large-Scale Automatic Classification of Phishing Pages.” In: *NDSS*. Vol. 10. 2010.

[26] The Free Encyclopedia Wikipedia. *bitly*. 2016. URL: `https://en.wikipedia.org/wiki/Bitly` (visited on 01/08/2016).

[27] The Free Encyclopedia Wikipedia. *URL shortening*. 2016. URL: `https://en.wikipedia.org/wiki/URL_shortening#History` (visited on 01/08/2016).

A All malware flagged hostnames

Below all hostnames for malware flagged sites are shown.

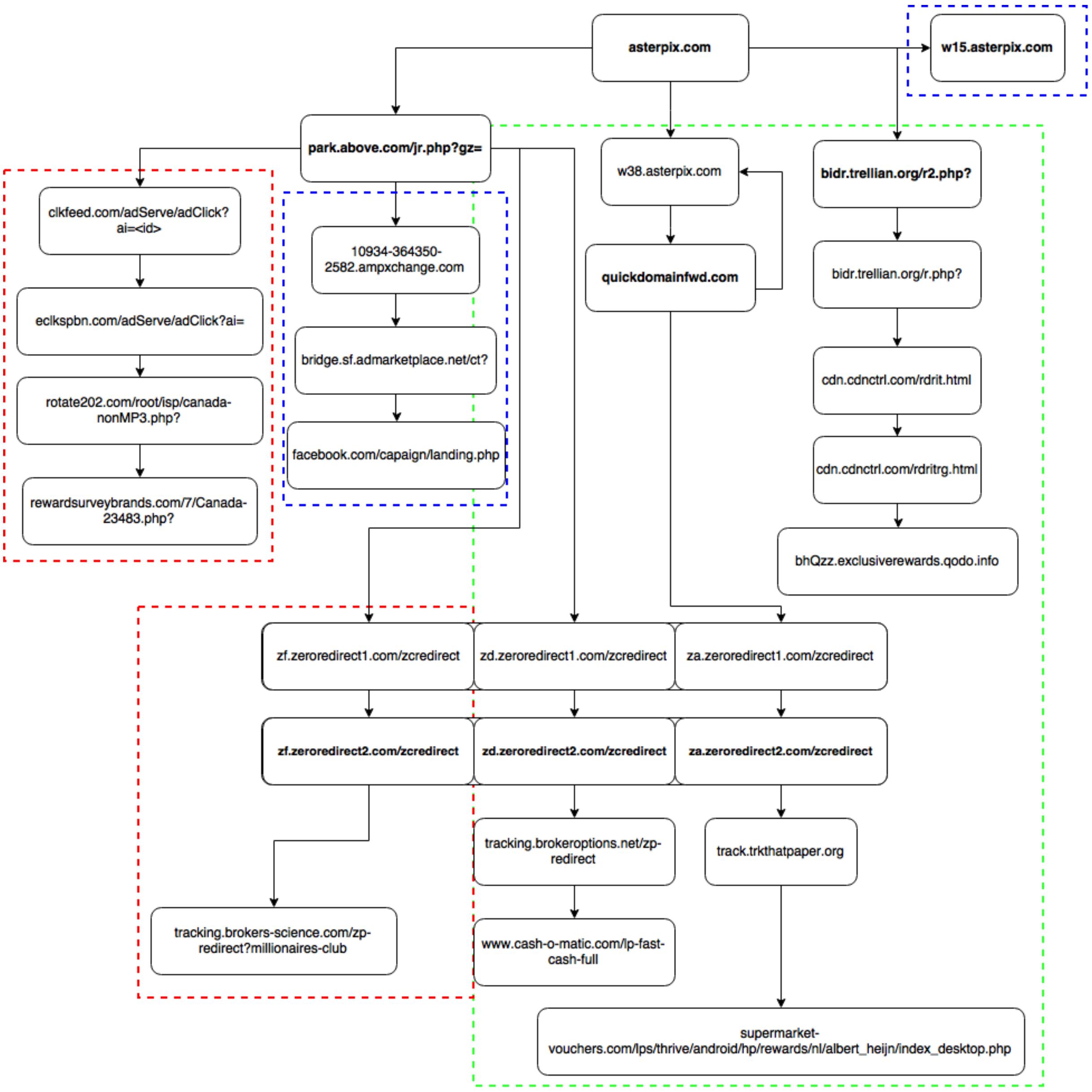
Hostname	**count**
www.asterpix.com	495
video.asterpix.com	113
www.tagvn.com	75
www.filelodge.com	57
keyknowhow.com	23
hurl.content.loudeye.com	16
static.zangocash.com	14
www.perfectporridge.com	13
www.content.loudeye.com	5
twitter-buzz.blogspot.com	4
www.mp3-host.com	4
tweetergetter.com	4
britanniaradio.blogspot.com	4
scripts.cgispy.com	3
www.verstuurpersbericht.nl	3
makelovereal.net	3
psfights.com	3
www.awflasher.com	3
www.yangss.com	3
meirixiaochao.com	2
www.tutorialhero.com	2
cbclickbank.com	2
www.myspace.com.on.nimp.org	2
fotos.nitosblog.com	2
www.psfights.com	2
tinypic.info	2
www.open.org	2
filmmakingcentral.com	2
www.hentai.net	2
loda.jp	2
forum.psfights.com	2
gw.tv	2
videodownloader.net	2
www.dark-wraith.com	1
www.brookview.karoo.net	1
gmail.sayni.net	1
firstpost.on.zoy.org	1
www.buyaggressive.com	1
www.incentaclick.com	1
www.cbclickbank.com	1
catholic-dads.blogspot.com	1
www.w70.on.nimp.org	1
anime.on.nimp.org	1
www.bagja.com	1
halflife2.zoy.org	1
eaglebird.on.nimp.org	1
www.sharebees.com	1
HTNL.ON.NIMP.ORG	1
www.steeg.co.uk	1
nightlife.sea.free.fr	1
www.flowers-for-all-occasions.co.uk	1
free.of.pl	1

Hostname	**count**
www.handbags-purses-guide.com	1
utopia.ision.nl	1
www.usasaler.com	1
www.megaresistencia.com	1
www.ADDZUP.com	1
defocus.on.nimp.org	1
www.startdisk.com	1
www.nn47.com	1
digdist.bbsindex.com	1
www.ragnarokonline.com	1
www.helpingpreneur.com	1
webdoispontozeroplus.blogspot.com	1
santhuongmai.com	1
iain.on.nimp.org	1
trap.on.nimp.org	1
www.vintagefair.co.uk	1
photos.verdebrick.com	1
forex.aminadab.com	1
www.componentspot.com	1
bbs.51hgame.com	1
www.realisticforeignpolicy.org	1
sureshkalmadi.org	1
www.webcultura.eu	1
veejays.porn.site.on.nimp.org	1
dark-wraith.com	1
www.gbfam.com	1
www.pierre-markuse.de	1
L33tmasta.on.nimp.org	1
www.betterdaystv.net	1
bybk.net	1
terenasstand.on.nimp.org	1
www.whymilk.com	1
WARM.on.nimp.org	1
www.mpegjoy.com	1
library.on.nimp.org	1
www.GNAA.on.nimp.org	1
sarahzfucked.on.nimp.org	1
ent.163.com	1
www.presidiacreative.com	1
news.on.nimp.org	1
torrentmatrix.com	1
www.tutorialized.com	1
IRChax.on.nimp.org	1
mrangle.macroviz.com	1
www.redsofts.com	1
www.lunarembassy.com	1
frontpage.on.nimp.org	1
egres.on.nimp.org	1
www.mannpill.com	1
www.simonemoticon.com	1
myspacepix.on.nimp.org	1
www.happydietnews.com	1
hahahahahahahayougotfuckingowned.on.nimp.org	1
www.santhuongmai.com	1
members.on.nimp.org	1

Hostname	count
www.tinypic.info	1

Table 6: All malware hits

B Asterpix Graph



asterpix.com
w15.asterpix.com
park.above.com/jr.php?gz=
w38.asterpix.com
bidr.trellian.org/r2.php?
clkfeed.com/adServe/adClick?ai=<id>
10934-364350-2582.ampxchange.com
quickdomainfwd.com
bidr.trellian.org/r.php?
eclkspbn.com/adServe/adClick?ai=
bridge.sf.admarketplace.net/ct?
cdn.cdnctrl.com/rdrit.html
rotate202.com/root/isp/canada-nonMP3.php?
facebook.com/capaign/landing.php
cdn.cdnctrl.com/rdritrg.html
rewardsurveybrands.com/7/Canada-23483.php?
bhQzz.exclusiverewards.qodo.info
zf.zeroredirect1.com/zcredirect
zd.zeroredirect1.com/zcredirect
za.zeroredirect1.com/zcredirect
zf.zeroredirect2.com/zcredirect
zd.zeroredirect2.com/zcredirect
za.zeroredirect2.com/zcredirect
tracking.brokeroptions.net/zp-redirect
track.trkthatpaper.org
tracking.brokers-science.com/zp-redirect?millionaires-club
www.cash-o-matic.com/lp-fast-cash-full
supermarket-vouchers.com/lps/thrive/android/hp/rewards/nl/albert_heijn/index_desktop.php

Figure 5: Asterpix Graph of 22-01-2016: The Green box encompasses, redirects and pages encountered on the European VPN. The red box contains the American redirects and landing pages. The blue box shows the redirects and pages from the Asian node.

C Dataset entry example

Below is an example of a malware hit in our dataset. The structure is in JSON.

```
{
        "_id" : ObjectId("569e8a025c59732e6d8a38c2"),
        "malware" : true,
        "service" : "http://bit.ly/",
        "short_hash" : "k8gnm",
        "short_url_metadata" : {
                "created_at" : "2009-08-20T11:14:54",
                "referrals" : [ ],
                "created_by" : "bitlyapidemo",
                "total_clicks" : 0
        },
        "malware_result" : {
                "phishtank" : false,
                "spamhaus" : [ ],
                "google" : [
                        "goog-malware-shavar"
                ]
        },
        "long_url_metadata" : {
                "header" : {
                        "date" : "Tue, 19 Jan 2016 18:54:27 GMT",
                        "last-modified" : null,
                        "content-length" : null,
                        "content-type" : "text/html"
                },
                "cert" : "",
                "script_links" : [ ],
                "alive" : true
        },
        "long_url" : "http://video.asterpix.com/v/9184721/torn-at-the-
           secret-policemans-ball-by-david-armand/",
        "service_id" : 1
}
```