A Measurement Study of Web Redirections in the Internet

by

Krishna B. Vangapandu

(Under the direction of Kang Li)

Abstract

The use of URL redirections has been recently studied to filter spam as email and web spammers use redirection to camouflage their web pages. However, many web sites employ redirection for legitimate reasons such as logging, load-balancing and security. While a majority of the studies on URL redirection focused on spam redirection we provide a holistic view of the use of URL redirections in the Internet. We performed a redirection study on various sets of URLs that includes known legitimate and spam websites. We observed that URL redirections are widely used in today's Internet with more than 40% of URLs in the top websites redirecting for various reasons. It was observed that redirections occur in chains whose length on an average was close to 1.4. We also observed that server-side redirection is prominent in both legitimate and spam redirections. Spam redirections employed JavaScript redirections more often and a very high percentage of spam redirections lead to an external domain. Apart from providing a quantitative view of URL redirections, we also provide a further classification of legitimate redirections based on the reason of redirection. We expect that our measurement results and classifications to provide a better understanding of the usage of URL redirection, which could help improve the spam filtering and other applications that rely on URLs as the web identifiers.

Index words: URL, JavaScript, Redirections, Spam, Web-Crawlers

A Measurement Study of Web Redirections in the Internet

by

Krishna B. Vangapandu

B.E., Osmania University, India, 2006

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Athens, Georgia

2008

A Measurement Study of Web Redirections in the Internet

by

Krishna B. Vangapandu

Approved:

Major Professor: Kang Li

Committee: John A. Miller
Shelby H. Funk

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2008

Dedication

To my mother Rama, my father Kersi; for all the love they showered on me, for supporting me in the hardest of times. To my uncle Noshir Sethna and brother Vamsi Krishna, my friends Sridhar and Ashish for their constant encouragement and support.

Acknowledgments

I would like to thank Dr. Kang Li, Dr. John Miller, Dr. Shelby Funk for their willingness to participate in the defense committee. Dr. Kang Li's invaluable guidance has been the driving force throughout the Master's Program. I would like to thank the Department of Computer Science at The University of Georgia, for the opportunity to pursue the Master's program and the assistantship they granted me during the course of the program. Finally, I would like to thank Noshir Sethna for proof-reading this document.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many popular Internet applications use URL as web object identifiers. For example, search engines use URLs in their indexing schemes [1]. Content blockers and blacklists often rely on URLs to filter unwanted content. However, URL redirections provide challenges to these applications [2, 3, 4]. URL redirection is a widely used technique to make a webpage available under different URLs. Spammers use redirections as means of cloaking to boost the spam pages' rankings at search engine and evade content filtering and blacklist.

Since redirection is one of the most widely used spam-hiding techniques, much research has focused on using URL redirections as an important factor in detecting spam. These previous works study URL redirections in the spam websites exclusively. However, many legitimate (non-spam) websites also employ redirection for reasons such as load balancing, link tracking, and bookmark preservation. To build accurate and effective spam detection based on URL redirections, we need to understand the use of URL redirections in both spam and legitimate sites.

The contribution this thesis makes is to provide a holistic view of URL redirections in the Internet. We measure redirections in both legitimate and spam websites in the current Internet to provide a comprehensive view of URL redirections. We classified redirections based on multiple aspects, including the types of techniques (Server-side, JavaScript and META redirection), target of redirection (internal vs. external redirections), and on the reasons that motivate these redirections. We expect that our measurement and classifications can provide a better understanding of the usage of URL redirection, which could help improving the spam filtering and other applications that rely on URLs as the web identifiers.

From our study, we observed that redirection is very common in both legitimate and spam websites. Legitimate websites (40.97%) redirect as much as the spam websites (43.63%). Among all known types of redirection techniques, server-side redirection is the most commonly used technique which accounts for 70-90% of legitimate redirections, and 70% of spam redirections in our datasets. Differing from legitimate URL redirection, JavaScript redirection is detected more often in redirections in spam websites (30%) than those used in legitimates web site (less than 10%).On further classification of redirections based on the URL target it was observed that the spam websites redirect to external servers more often than the legitimate websites. Over 46% of spam redirections are external while only 2% of the legitimate redirections such as those in the Alexa Top Websites were external, stating that the higher the probability of the URL being legitimate lesser the chances of it redirecting to an external domain.

Our results also provide a quantitative analysis of common reasons for using redirection in legitimate sites. Among the redirections that stay within the domain, the dominant reasons are virtual hosting and load balancing; whereas, for redirection to external sites, the dominant reasons are link tracking for advertisement and providing warnings.

Chapter 2

Understanding URL Redirections

This chapter explains fundamental concepts that the thesis is based upon. It aims to provide a detailed understanding of various terms and technologies that are the basis for the work.

URL (Uniform Resource Locator)

A URL is a string that is used to identify a resource available within a network. URL is heavily used in the Internet to identify resources and also provides a means to locate the resources by describing its network location. Generally, a web URL is of the format: `(protocol)://(server)/(resource path on the server)`. The protocol in case of web URL is usually http or https. URL is the most important element of this study and we primarily deal with the situation where a resource has more than one URL as identifier, since the resource is no longer uniquely identified.

URL Redirection

URL Redirection is said to occur when a request results in an additional request to a resource without user intervention. Simply put, a URL is said to be redirected, if a client requests a resource located at a specific URL, but the client's final destination at the end of the request is a different URL.

2.1 Types of Redirection Techniques

This section describes the types of redirection techniques as well as the common reasons for redirections. Based on the implementation techniques, URL redirections are classified into

1. Server-side redirections,
2. JavaScript redirections,
3. META redirections.

2.1.1 Server-Side Redirections

Server-side redirection occurs when a client requests a resource and the server issues a directive in the form of HTTP status codes which makes the client request through a different URL. HTTP reply status codes of type 3xx as well as some 4xx with a location field in the header imply that the client has to redirect to a different URL. For example, request for http://www.google.net returns a status code 302 redirecting the request to http://www.google.com.

Redirections can also be performed by using publicly available services redirection services (such as tinyurl, shorturl). It was also observed that some other services are exploited by spammers to perform redirection even though they are not exactly meant to generate redirections. An example of this service is the "Google, I'm feeling lucky" which automatically redirects the user to the first result for a query made. When the user clicks on the "I'm feeling lucky" button to submit a search query, it attaches an additional parameter "btnI" to the query string that is submitted to the server. The server then redirects the user to the first result for that query. Since this is done via URL through query string parameters, one could use this as a redirection service and the URL would still appear as if the content is hosted on Google.

2.1.2 JavaScript Redirections

JavaScript redirections are triggered through statements like "window.location=someurl". Such instructions are inserted into the script sections of the HTML and when the client JavaScript engine executes these statements, the engine instructs the browser to redirect to the URL specified within the script. Unlike server-side redirections, a web page is actually loaded partially or completely into the client browser before the redirection occurs.

Rich Interactive Components that the browser host are developed using Flash, Silverlight, AJAX techniques and to some extent Java applets. These components interact with the browser through native browser interfaces and can initiate redirection of the user to a different URL by accessing the browser DOM. Thereby these redirections are translated to JavaScript calls and thus classified under JavaScript redirections.

2.1.3 META Redirections

Another client side redirection technique makes use of META tags located in the HEAD section of the HTML page. By setting the associated "http-equiv" attribute to "refresh" and the "content" attribute to a target URL, a redirection would be triggered at the client browser to this target URL. The redirection happens after a browser finishes parsing a HTML page, then the META refresh action is triggered to load content from the target URL.

2.2 Related Work

Most of the previous work in the field of web redirections focused on spam redirections. In one of the early works on web spam classification, Gyngyi and Garcia-Molina [2] describe redirection as a spam hiding technique used by spammers to create doorway pages. Based on a quantitative preliminary study Wu and Davison [3] found the presence of cloaking in the Internet. They looked at redirections as one of the techniques used to perform cloaking. The current study differs from these by considering not only spam datasets but also a few common categories of legitimate web sites.

JavaScript redirections have been shown to be used by spammers as a way to dupe users into viewing spam. Benczr et al. [4] discovered numerous doorway pages which rely on JavaScript redirection. Chellapilla and Maykov [5] looked at JavaScript redirection explicitly with a focus on the techniques employed by the spammers. The Microsoft Strider team in their work on systematic discovery of spammers emphasized URL redirection as a common spam technique. They developed a tool, Strider URL Tracer, which can be used to detect all the domains that a current web page connects to. With the aid of the Strider, Wang et al. [6] studied URL redirections in the context that there are content providers which redirect the user to malicious sites. Niu et al. [7] conducted a study on forum spamming with context-based analysis using the Strider to identify doorway pages.

To our knowledge, most of the work mentioned above studied redirection in the context of spam. The approach of the current study is different from the work mentioned above as it looks at general web redirections as a whole. Our study involves detection of URL redirection, classification of detected redirections across multiple dimensions.

2.3 Reasons For Using Redirections

As mentioned earlier, there are several reasons for URL redirection - both legitimate and illegitimate. In this section we look at some of the most common reasons for employing redirection. The order of appearance is based on the popularity among the classification results with most commonly observed reason listed first.

2.3.1 Virtual Hosting And DNS Aliasing

Virtual Hosting and DNS Aliasing are where more than one site (or domain name) is mapped to a single IP address. This is commonly used by websites to register most commonly misspelled domains and redirect requests to these domains to the original server. Such organizations register the same domain name with different top-level domains. For example, requests to gooogle.com or google.net all redirect the user to http://www.google.com.

2.3.2 Load Balancing

Most of the top websites host content on several servers. These servers either host specialized content or mirror each other. In cases of high volume of web traffic, requests are redirected to one of these servers. The criteria of these redirections are dependant on the website. For example, popular websites like search engines host different mirror servers and requests are redirected based on the nature of resource requested.

2.3.3 Link Tracking (via indirections)

Many websites use redirection for statistical and logging reasons. For example, websites log the advertisement clicks before it actually takes the user to the advertised webpage. For this to happen, advertisement clicks are taken to the originating website where the information is logged and then the request is redirected to the advertised webpage.

2.3.4 Resisting Web Spam (via indirections)

Many websites rewrite external links in their web pages by introducing a level of indirection through a server that is not indexed by search engines. For example, all user links posted at MySpace are disguised as a link from the domain name msplinks.com, not myspace.com. A web page linked from the former domain is much less valuable than a link from the latter one.

2.3.5 Providing Warnings (via indirections)

A level of indirection is also used to provide a warning to users when they are about to leave the current domain. Most websites introduce such a redirection in order to waive responsibility of the content that the subsequent pages present.

2.3.6 Security

Unauthenticated or unauthorized requests to a resource are usually redirected to a login page or an information page. For example, visiting a protected profile on MySpace redirects unauthenticated users to the login page. Similarly, transition from HTTP to HTTPS is often enabled through redirection. For example, a request to http://www.bankofamerica.com would be redirected to https://www.bankofamerica.com/index.jsp.

2.3.7 URL Rewriting

Some websites rewrite long URLs with short and user-friendly URLs. When a request is made with these shorter URL, a redirection to the actual URL occurs. Many redirection services such as tinyurl rely on redirections to provide short and clean URLs.

Redirection is also used for many other reasons, such as to keep bookmarks, to route requests in CDN, to redirect requests for invalid resources, and to provide personalized content based on client geo-locations or user-agent types. For example, we observed that 15 out of Alexa's top 500 websites redirected differently when the "user-agent" field is changed from a popular browser (Firefox) and a crawler (Google-bot).

Note that in the classification of observed redirections, it is very unlikely to distinguish between URL rewriting, homepage navigation, DNS aliasing and Virtual hosting using URL heuristics with high accuracy and thereby all these three have been singularly classified under Virtual hosting. Even though URL redirection is one of the most exploited techniques for spamming, the above mentioned reasons justify its legitimate presence.

2.4 Classifying Redirections

This section describes this study's efforts at classifying URL redirections into several categories based on the implementation techniques (Server-side, JavaScript, or META), the target of redirections (external vs. internal), and the reasons for redirection.

2.4.1 Using Redirection Techniques

Classification based on the type of redirection techniques helps us identify the most commonly used techniques in various types of web sites. We briefly overview the required efforts to detect and classify redirections based on their implementation techniques.

The detection of redirections can be achieved with or without a browser. For example, the Microsoft Strider URL Tracer is useful in detecting redirections from a URL by loading the page into Internet Explorer. While detecting server-side and META redirections is relatively simple, JavaScript redirections are complicated to detect. Standalone JavaScript interpreters, such as the Rhino JavaScript engine [8], have been used but they do not capture all intended JavaScript executions. Browser based tools helps us detect JavaScript redirections with high accuracy but these tools have high overhead in the form of CPU consumption by the graphics rendering engines. Non-browser based tools can process at high rate but incur significant false negatives in detection of client side redirections. Thus in order to achieve faster and accurate results, both these methods should be used wherever applicable. For example, non-browser based methods are generally sufficient to work with server-side or META redirections.

A. Detecting Server-side Redirections

The detection of Server-side redirections can be easily achieved by monitoring the status code returned in HTTP responses. Browser performs a redirect when the status code returned in the response is of type 3xx or 4xx with the "location" property set in the response to a URL.

B. Detecting META Redirections

META redirections can be detected by looking for META tags with the attribute "http-equiv" set to refresh and the "content" attribute set to a different URL. An example of such a META tag is shown below:

```
<META content="2;url=http://uga.edu" http-equiv="refresh">
```

C. Detecting JavaScript Redirections

The detection of JavaScript redirections is not straightforward because of the possible level of obfuscation [5]. For example, shown below is a simple JavaScript statement that initiates redirection:

window.location=”http://myspamlink.com”

Given the amount of flexibility that JavaScript offers to the web developers, the same statement can be obfuscated to many different versions that are hard to parse without executing the script. For example, the above mentioned script could be written as

eval(”m̈oc.knilmapsym//:ptth=̈ noitacol.wodniw” .split(””).reverse().join(””));

Thus, a detector should be capable of executing JavaScript and using a web browser is the best way in which one could detect all complex redirections. Moreover, rich interactive controls such as Flash embedded within the page can also trigger redirections. It is a very tedious job for non-browser based detectors to capture such redirections and in case of browser-based detectors, one should have complete control over the browser interactions which modern browsers do not allow. Thereby, such redirections are classified as JavaScript redirections.

More details on how redirections can be detected and Java code to perform the detection is provided in the Appendix A.

2.4.2 Using Redirection Targets

Redirection target can be a URL either in the same (internal) domain or external domain. The classification of redirections into internal and external targets is used to validate a commonly held hypothesis that redirections at spam websites often have more external redirections than the redirections used in legitimate sites. The distinction between internal and external redirection is defined by the web domain ownership of the original and the target

URL. An external redirection is defined as a redirection between two URL domains which are not owned or managed by the same organization.

Detection of external redirections is not straight-forward. Very often different domains are managed by the same organization. For example, a redirection from the msdn.com domain to the microsoft.com is not considered as external redirection since both these websites are managed by the same organization. Unfortunately, there are no systematic methods to check if two sites are owned by the same organization because of which we adopt the following heuristics to differentiate internal redirections and external redirections. For a redirection from the original URL domain X to a target domain Y, we check:

1. If the domain X is a sub-domain of Y
2. If they share a common domain name server
3. If they are two domains with a common top-level domain
4. If their IP address is in the same Class B range

If any of these checks returns a positive result, it may be considered that X and Y belong to the same organization, thereby classifying the target as internal. The above described heuristic has been validated with a set of known inputs and the classification accuracy was observed to be high(upto 80%). This classification is essential since the redirection targets play a key role in understanding the reasons behind redirection.

2.4.3 Using Reasons For Redirection

Classifying redirections based on reasons for redirection gives us an in-depth understanding of the usage of URL redirections in various web sites. Since there is no precise information to determine the reason for using redirections, the following methods are used to infer and classify them into categories. First, a manual inspection of a small set of URL redirections in the legitimate dataset is conducted to infer the motivations of redirections based on the

naming conventions of original and target URL. Second step involves extraction of keywords that may represent a specific reason for redirection and the heuristic classifier is built based on these keywords. For example, if keywords such as "login," "auth," "https," appear in the destination URL, then the redirection is classified under "Security"' category. Keywords such as "ad," "click," "overture," and "adtmt" represents advertisements. We apply these heuristics with the classifier to the rest of the URL redirections and randomly sample the outcome for manual verification of the classification results. Throughout the study, the keywords-based URL heuristics are applied to identify redirections that are for link tracking, security, directory listings, etc.

Apart from the keyword heuristics, we also use other techniques to identify motivations such as load balancing and virtual hosting. These are determined by matching the domain names in the target and destination URL. For example, there is a close match between the domain www0.shopping.com and www1.shopping.com. The difference lies in the first part of the domain name with indications of a possible use of load balancing servers.

To justify the accuracy the classification results between reasons-heuristics and target-heuristics were compared. Some of the reasons such as malware are highly unlikely to appear in the redirections of Alexa Top 500 websites and the results we present in the next section demonstrate this assumption. Consider the case of Alexa dataset, where the reasons-heuristics resulted in 0% of external redirections having reasons such as Load Balancing and Virtual Hosting which can be justified with the common notion of these top Internet organizations hosting their own web farms. As another justification, it was observed that none of the URLs that were detected to have internal redirection were found to redirect for malware reasons.

Chapter 3

Experiments And Results

This chapter describes the system used to detect redirections and the datasets used in this study. The first section describes the system that has been used to study redirections followed by the section that explains the datasets that were used for the study. Finally the results obtained are explained in detail.

3.1 System Description

The system used to study redirections is comprised of a custom crawler to collect the datasets, a redirection detector, an external redirection classifier and a heuristics-based reasons classifier.

The redirection detector uses SWT Browser Widget [8], a browser component that is commonly used in Java-enabled applications. For example, Eclipse, a popular Java IDE, uses SWT Browser component in its in internal web browser. Though the SWT Browser is a GUI widget, the redirection does not use a graphical user interface and thus does not require user interaction. Because the system uses a real browser, it can accurately detect all known types of redirections. The redirection detector parses each URL and redirections detected are classified as a server-side or client-side redirections. The URLs containing client-side redirections are further parsed and classified into JavaScript or META redirections. The system also contains a component that can be used to detect external redirections. The internal and external redirections are further processed by a classifier component which uses URL heuristics to logically classify the redirection based on the motivation.

3.2 Datasets

In order to study the prominence of redirections different types of datasets were considered. Most of the study focused on the legitimate websites and as a reference a small spam dataset whose URLs are taken from spam honeypots is included. The spam dataset is used so as to provide an approximate comparison between legitimate and spam redirections. The legitimate datasets used are described in subsequent sections.

3.2.1 Alexa Websites

Alexa is one of the top sources for web traffic information. It collects web crawls and information related to Internet traffic using toolbars. As a portion of the information that it provides, Alexa lists the top 500 websites in the Internet. This dataset is crucial since this represents the most heavily used websites in the Internet. The redirections detected in this dataset would reflect the day-to-day redirections experienced by the users.

3.2.2 University Websites

The second dataset is comprises of the Top 500 University websites whose list is obtained from arwu.org. This website is organized by Institute of Higher Education, Shanghai Jiao Tong University and is one of the prominent sources of university rankings. The motivation behind choosing this dataset is the belief that the nature and design of university websites is completely different from the commercial websites like those of Alexa dataset. Most of the pages in the Alexa's top websites are dynamic while the majority of university websites host static content.

3.2.3 Top Keywords

Another dataset that was used in the study is prepared from the websites returned by the Google search engine for the Top 500 keywords used in a timespan of 90 days (see Appendix C). While Alexa ranked websites are the most consistently visited websites, this dataset

Table 3.1: Summary of Redirection Detection Results

Dataset	URLs	Total Redirections
Alexa Websites	107300	40.97%
Top Universities	88567	18.75%
Top Keywords	99389	20.81%
Spam Dataset	13451	43.63%

comprises of links for the keywords that were most queried on the Google search engine by world-wide users. This dataset cannot be accurately classified as legitimate dataset since most spammers target these keywords and there is a high possibility of having spam websites returned by the search engine.

3.3 Results

The results obtained by conducting experiments on the above mentioned datasets are explained in subsequent sections. The amount of redirections observed in the datasets was observed to be significantly high in the Alexa dataset (40%) and spam dataset (43%) while in the rest of the datasets it was observed to be around 20%.

3.3.1 Overview Of Redirection Measurements

The numbers of URLs in each dataset as well as the total number of redirections (in percentage) are listed in Table 3.1. Figure 3.1 provides a further breakdown of these results based on the techniques used to implement redirections.

Overall, URL redirection was observed to be common in all forms of websites irrespective of the nature of the data set. It was observed that server-side redirections are predominant. These observations are true for popular Internet sites as indicated by the Alexa dataset, as well as the other datasets. About 25-40% of legitimate URLs actually involve redirection

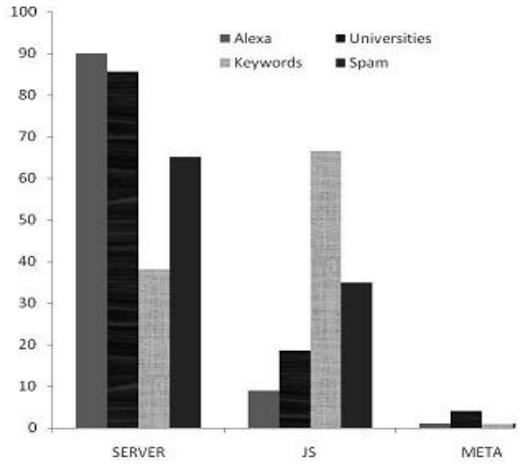

Figure 3.1: URL Redirection Comparison

and the observed redirections are mostly server-side redirections. Study on completely spam websites presents a similar result: overall 43.63% of the URL in the Spam dataset redirected to a different location in which server-side redirection (69.33%) is observed the most.

The keywords dataset has the highest percentage of JavaScript redirections while the Alexa dataset has the lowest. The top websites as in the Alexa dataset tend to rely on server side redirections to maintain the load on the websites.

Some noteworthy observations in META redirections was that the university dataset has a relatively higher percentage of META redirection. META redirection is most likely used when the webmaster does not have control over the server which is common with the university hosted web pages. Given detection of META redirections is straight-forward, it makes sense that spammers do not employ this technique often and only 0.46% of the spam dataset redirected using META techniques.

Figure 3.2 shows the breakdown of redirection types for one the early datasets we considered. The datasets used at that time consists of blogs and websites hosted by the University of Georgia. It can be observed that these results differ from the originially shown results

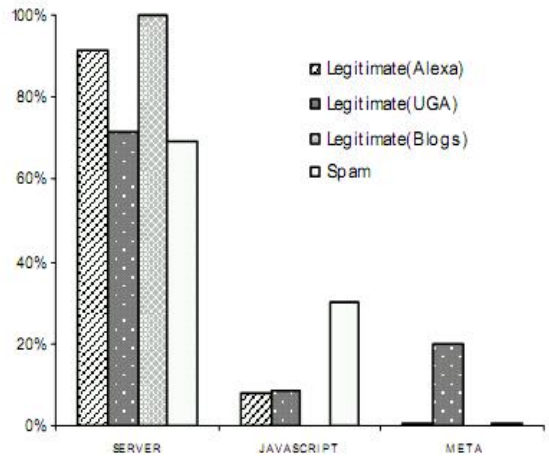

Figure 3.2: Initial results for URL Redirection Comparison

vary in the amount of JavaScript and META redirections. This gives an impression that JavaScript redirections (combined with external redirection detection) has high probability of being spam. But having conducted the experiments with the final datasets, the idea of using JavaScript redirections as a measure to detect spam no longer holds good.

3.3.2 Impact Of Popularity And Depth

While the overall results demonstrate the wide use of redirection in legitimate sites, it is still interesting to see whether the use of redirections have any direct connections with the popularity of a website or with the depth of the URL on a web site. It can be expected that the legitimate redirections should not be effected by the popularity of a website since the requirements to perform load balancing, virtual hosting or for other reasons be the same across all major websites irrespective of its popularity. Since search engines rank redirection URLs at a slower rate than the ones without redirections, redirections at the top levels should be less than those in the next levels. To study the impact of popularity Alexa top 500 websites were studied for redirections. For the study on impact of depth, Alexa top 100

Table 3.2: Impact of Popularity on Redirections : Alexa Top 500 Level-1 Pages

Dataset	**% Redirections**
Alexa Top 100	18.00%
Alexa Top 101-200	17.00%
Alexa Top 201-300	22.00%
Alexa Top 301-400	23.00%
Alexa Top 401-500	17.00%

Table 3.3: Impact of Depth on Redirection in Alexa Top 100(upto 3-levels)

Dataset	**% Redirections**
Alexa Top 100 Level	% Redirections
Level 1	18.00%
Level 2	38.05%
Level 3	41.47%

websites were crawled upto 3 levels. Baeza-Yates et al. [10] found that crawling websites up to 3-5 levels is sufficient to cover over 90% of the pages linked.

The breakdown of percentage of redirections observed across the Alexa Top 500 websites (Level-1 pages) is shown in the Table 3.2. From the results, it was observed that the popularity of a website has no significant impact on the amount of redirection observed. In spite of 2-3% variation, each sector is close to the average amount of redirection (19.40%). The result observed confirms our expectation that amount of redirections does not impact the popularity of a website (or vice-versa).

Results from experiments on the first three levels of Alexa Top 100 Websites is shown in the Table 3.3. It shows that the amount of redirection increases with depth, i.e, there is a direct correlation of amount of redirections and depth.

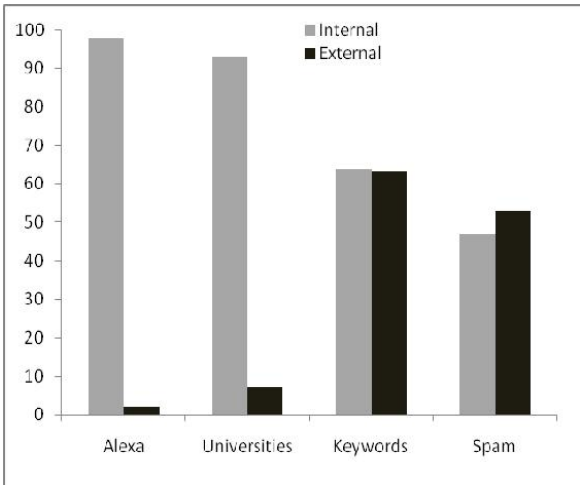

Figure 3.3: Comparing Internal and External Redirections

3.3.3 Comparing External And Internal Redirections

Classification of the redirections into internal and external redirections gives a deeper insight behind the reasons for using redirections. We expect the URL from Alexa dataset to have more internal redirections while those URL returned from the spam dataset to have more external redirections. Websites prefer to redirect from an internal URL to external webpage for different reasons. Adding a level of indirection often allows websites to track the links that are leaving their website or add an additional layer of protection for the users. A look at the detailed classification results in the subsequent sections provides a better understanding as to why internal or external redirections take place.

Figure 3.3 shows a graphical comparison between the amount of internal and external redirections in the datasets. The percentages may not add upto 100 because the value is computed agains the total number of URL that has one or more redirections. Had the value been calculated against the total number of redirections (which would be close to the total number of URL that were redirection multiplied by the average chain-length), the values

would add upto 100. This is because a URL can have multiple redirections which can be a mixture of internal and external redirections.

The Alexa dataset has the least percentage of external redirections while the keywords and spam datasets have higher percentages of external redirections. This meets our expectation about redirections in Alexa dataset since not many popular websites would want to divert their traffic to external websites unless the redirections are for advertisements which is the major reason for external redirections. The nature of university dataset with respect to internal and external redirection is similar to that of the Alexa dataset and the nature of keywords dataset is similar to the spam dataset. There is a high possibility that quiet a lot of spam websites are present in the keywords dataset since the goal of spammers is to optimize their pages such that they appear high on the search results. Though this assumption has not been verified in the dataset, there have been several keywords that were spam-prone. Subsequent sections present a refined classification of these redirections based on the reasons.

3.3.4 Multiple Redirections And Redirection Chain Length

Redirections can occur in chains where there are multiple redirections occuring before the final URL is reached. Experiments showed that more than one redirection is very commonly observed. In case of multiple redirections, over 4% of the URL in all the datasets had both JavaScript and Server-side redirections. We expect the maximum chain length to be upto 5 since it is the recommended maximum by the World-Wide Web Consortium(http://www.w3c.org) to be used in the web clients such as a Web browser. Though the average chain length was observed to be close to 1.5 after redirection chains were filtered to remove duplicate redirections, redirection chains with lengths of over 100 was commonly observed.

A simple scenario for this would be the case where a server is configured to redirect to an information page anytime a request results in the status code 404 (Page not found) and the information page itself is missing on the server. It was also observed that some of the

requests made using non-browser tools, with no user-agent set, would result in redirection chain lengths of over 500 while the same request when made from a browser results in no redirection. This behavior can be attributed to the non-standard nature of browsers which force the web administrators to configure the web server to serve different content based on the type of the user-agent that arrived in the request. Simple test on user-agent cloaking on Alexa Top 500 websites revealed that over 15 websites served different content for different user-agents. Modern browsers are aware of these long chain-lengths and thereby limit the number of redirections they follow. Mozilla Firefox, for example, by default limits the redirections to 20 and this limit is only applicable to the server side redirects and not to those of client side redirects. In the subsequent section we explain the results obtained by classifying the redirections based on reasons.

3.3.5 Classification Based On The Reasons For Redirection

In this section we further classify the internal and external redirections based on the reasons for redirection. As mentioned, URL heuristics are applied in order to perform these classifications. See Appendix B for more information on the heuristics and techniques that were applied. We expect the predominant reason for internal redirections to be virtual hosting and load balancing since these are the most common requirements for popular websites. We also expect advertisements to account for the major share of external redirections since it is the revenue generating source for websites. Comparing the expected results with those obtained, it can be strongly suggested that the heuristics have been accurate and that the applied heuristics would classify up to 80% of the URLs given. In the graphs shown in the subsequent sections, reasons whose count was found to be zero after the classification has been eliminated in the graph for the individual dataset so as to reduce noise on the horizontal axis. For example, the classifier detected no malware redirections in the Alexa dataset and it would not appear in the classification.

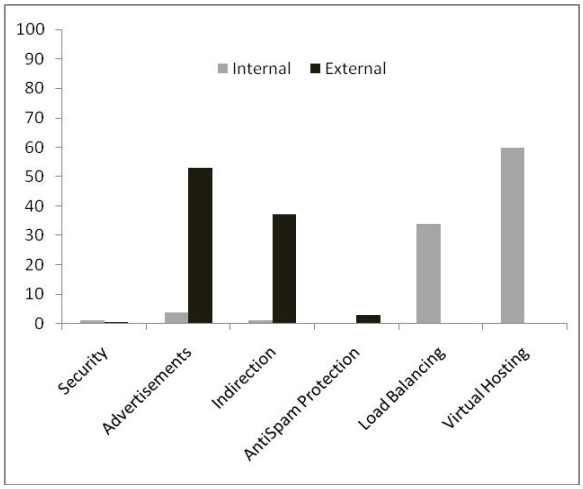

Figure 3.4: Classification of redirections in Alexa Dataset

Figure 3.4 shows the classification results for the redirections observed in the Alexa dataset. It can be seen that predominant reason for internal redirections in the Alexa dataset is Virtual hosting. Majority of the websites in this dataset rewrites URL and perform default homepage redirection which is classified under virtual hosting since there seems to be no easy way to distinguish between virtual hosting, URL rewriting or default homepage redirections. At the same time. these websites have high user traffic raising the need for load balancing because of which load balancing was observed to account for a significant portion(34.07%) of internal redirections. In case of the external redirections, advertisments(53%) and link-tracking(37.3%) holds a major share.

Performing a similar classification on the universities dataset tells us that major contributor for internal redirection is again load balancing followed by virtual hosting, the results of which are shown in the Figure 3.5. Security implementation in university websites is also a major contributor of internal redirections and it is more common for university websites to have a secured page than to display advertisements, because of which redirections for advertisements is observed to be very low. Few instances of malware redirections have also been

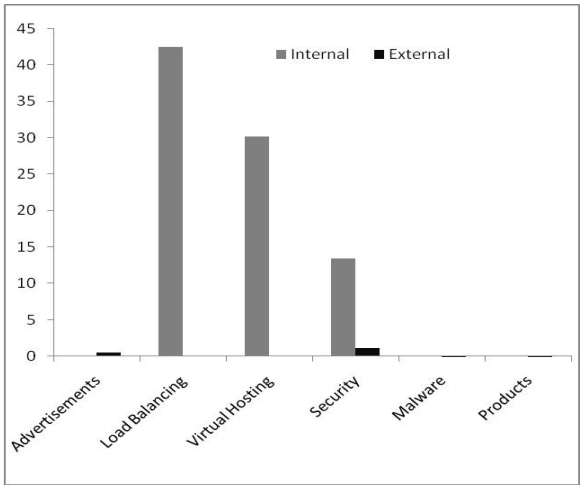

Figure 3.5: Classification of redirections in Universities Dataset

detected. Notice that the websites in Alexa did not have any instance of malware redirections in the results observed, since top websites usually invest heavily in making their websites safe to browse and are not spam-prone. In most cases, malware injections from these pages are removed very early on.

The results obtained by classifying the Keywords dataset are shown in the Figure 3.6. As mentined previously, the keywords dataset has a higher percentages of external redirections(63.3%) and advertisements(34.3%) appear to be the most common reason. Another noteworthy aspect is that the amount of malware redirections(3%) detected in the dataset strengthens the observation that there have been significant percentage of spam URL in the dataset collected. Internal redirections in the keywords dataset are primarily for load balancing(15%), virtual hosting(11%) and security(8%) reasons. Though not major, redirections to information pages(1%) as well as usage of redirection services(1%) have also been observed.

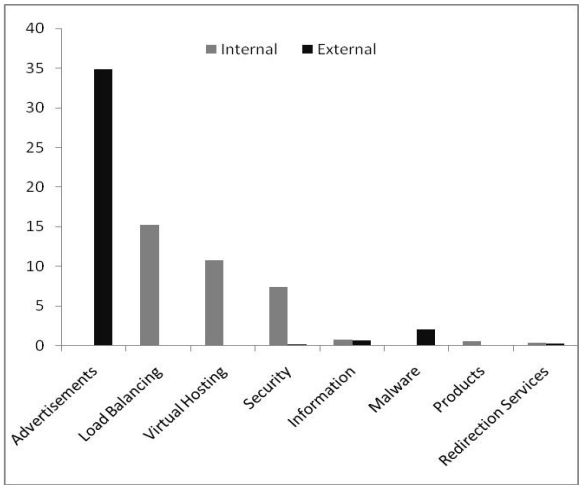

Figure 3.6: Classification of redirections in Keywords Dataset

The noteworthy observations made with respect to the keywords dataset are:

1. JavaScript redirections were observed to be the most dominant form of redirection in both the Keywords and Spam datasets.

2. The percentage of external redirections in both the datasets is higher than that of the internal redirections.

3. The percentage of malware redirections observed was significant in the keywords dataset and legitimate websites do not usually host malware. Though it has not been statistically captured, the amount of popup observed during experimenting with keywords dataset was higher than those observed in any datasets which is another indicator of malicious behavior.

Chapter 4

Conclusion

The results indicate that URL redirection is highly prevalent irrespective of the nature of the webpage (spam or non-spam, popular sites or university sites, static web or dynamic web content). These results indicate that the behavior of redirection cannot be strongly tied to Spam and that redirection is as significantly common in legitimate websites as in the spam websites. In fact a large variety of reasons for legitimate sites to adopt URL redirections were observed and statistically presented - some of the dominant reasons being virtual hosting, load balancing, and indirection provision for the purpose of tracking advertisements, etc.

One observation is that the JavaScript redirections are more prominent in the spam websites and keywords dataset that was prepared from the results returned from Google search for the top 500 keywords. In addition, many redirections in both the datasets led to external URLs, where a large majority of legitimate redirections are within the same domain. Since the classification is based upon heuristics, there is a minor error induced and thereby scope for improvement in heuristics may be necessary. It is also important to note that based on the nature of the datasets few of the heuristics may require further improvements so that the classification rate be improved. Heuristics for differentiating virtual hosts, URL rewrites would further improve the accuracy of the classification.

It is expected that the results of this study may benefit the applications that can potentially affected by URL redirections, such as search engine and content filter. Since the nature of Internet changes with time, the results might vary slightly based on the time of experiments. For example, one would probably find more redirections for advertisements, thereby more external redirections during holiday seasons. At the same time, we expect the variation

not to be significantly high since the requirements for load balancing, virtual hosting, URL rewriting are always present and currently URL redirection is the best technique that can fulfil these requirements.

Bibliography

[1] S. Brin and L. Page, The anatomy of a large scale hypertextual web search engine, in: Proc. 7th WWW, 1998.

[2] Z. Gyngyi and H. Garcia-Molina (2005), Web spam taxonomy, First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Japan, 2005.

[3] B. Wu and B. D. Davison (2005) Cloaking and Redirection: A Preliminary Study, First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb),Chiba, Japan, 2005.

[4] Benczr, K. Csalogny, T. Sarls, M. Uher, SpamRank - Fully Automatic Link Spam Detection, First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Chiba, Japan, 2005.

[5] Kumar Chellapilla, Alexey Maykov, Microsoft Live Labs. A Taxonomy of JavaScript Redirection Spam, ACM International Conference Proceedings Series; Vol. 215.

[6] Wang, Y. M., Beck, D., Jiang, X., Roussev, R., Verbowski, C., Chen, S., and King, S. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. In Proc. Network and Distributed System Security (NDSS) Symposium, February 2006.

[7] Y. Niu, Y. Wang, H. Chen, M. Ma, and F. Hsu, A Quantitative Study of Forum Spamming Using Context based Analysis," Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February, 2007.

[8] Mozilla Rhino - JavaScript Engine (Java). http://www.mozilla.org/rhino/

[9] SWT Browser Widget, Viewing HTML Pages with SWT Browser widget. Online at "http://eclipse.org/articles/Article-SWT-browser-widget/browser.html.

[10] Ricardo Baeza-Yates and Carlos Castillo, Crawling the infinite Web: five levels are enough. In Proceedings of the 3rd Workshop on Web Graphs (WAW), volume 3243 of Lecture Notes in Computer Science, pages 156-167, Rome, Italy, October 2004. Springer.

[11] Ben Feinstein and Daniel Peck, SecureWorks. Caffeine Monkey: Automated Collection, Detection and Analysis of Malicious JavaScript, BlackHat USA 2007.

[12] N. McFarlane. Fixing Web Sites with Greasemonkey. Linux Journal, Vol 2005, No. 138, 2005.

Appendix A

Redirection detection

In this chapter detection of Server-side, META and JavaScript redirections is described in detail. Detection of server-side and META redirections uses non-browser based methods where server-side redirections are detected using Java SDK network library and META redirections are detected using HtmlUnit, a test framework for web pages. JavaScript redirections are complicated to detect and requires a DOM-aware JavaScript engine and after evaluating various options, SWT Browser widget is used to load the URL and detect JavaScript redirections.

A.1 Detecting Server-side redirections

As mentioned earlier, server-side redirections for a web request made can be detected by monitoring the status code returned in the response. We track the status-code to see if it is of the type 3xx. The table A.1 presents a list of redirection 3xx status codes.

Figure A.1 shows the Java code that can be used to detect server-side redirections. By default, the HTTP client libraries provided in the Java SDK (or other API) is set to automatically follow server-side redirections, if any. If multiple redirections are to be tracked, the client library should be instructed to not follow redirections automatically. For example, in the code shown the method `setInstanceFollowRedirects(boolean yesNo)` in the `HttpURLConnection` class is used to disable the client from following the redirections automatically. The `HttpURLConnection` class provides methods which allows the developers set the timeout periods for the request made. Connection timeout can be set using the method `setConnectTimeout(int milliseconds)` and the read timeout can be set using the method

```
/**
 * Detects the Server Side Redirection Chains and updates accordingly.
 */
private void detectSSR() {
      finalUrl = cu.getMainUrl(); // get the URL of interest
      // detect server side redirections and count the redirections.
      while (true) {
        try {
           URL main = new URL(finalUrl);
           HttpURLConnection con = (HttpURLConnection) main
                                  .openConnection();
           con.setDoOutput(true);
           // disable automatic redirects
           con.setInstanceFollowRedirects(false);
           // set the timeouts
           con.setConnectTimeout(2000);
           con.setReadTimeout(2000);
           // establish the connection
           con.connect();
           // track the response code.
           int code = con.getResponseCode();
           if (code >= 300 && code <= 399 || code == 404) {
               finalUrl = con.getHeaderField("location");
               cu.RedirectionUrl.add(finalUrl);
               // increment the chain length
               cu.setChainLength(cu.getChainLength() + 1);
               // set Server Side Redirection detected as true.
               cu.setHasServerRedirection(true);
           } else
               break;
            } catch (Exception e) {break;}
        }
}
```

Figure A.1: Java code to detect server-side redirections

Table A.1: Redirection 3xx Status Codes

Status Code	% Description
300	Multiple choices
301	Moved Permanently
302	Found
303	See Other
304	Note Modified
305	Use Proxy
306	(Unused & Reserved)
307	Temporary Redirect

`setReadTimeout(int milliseconds)`. The default value for connection timeout and read timeout is 0 which indicates infinite timeout value. Setting the timeout explicitly prevents blocking of execution until the client is successful to connect and read the response. This is important to us since we perform detection on a large scale and an infinite timeout can block the execution.

A.2 Detecting META Redirections

META redirections can be monitored by looking at the HTML response returned for a URL. A META tag can instruct a redirection if its "http-equiv" attribute is set to "refresh" and the "content" attribute is set to a value of form "time-to-wait;URL" where URL is not the same as that of the current page. Shown below is a META tag that performs a redirection 2 seconds after the page is loaded, if some-url is not the same as the current URL with which the page has been retrieved.

```
<META content="2;url=some-url" http-equiv="refresh">
```

As mentioned previously, detecting META redirections is straight-forward. In our experiments, we made use of HtmlUnit(http://htmlunit.sourceforge.net) which is used to automate unit testing of websites. Figure A.2 shows the code that is used to detect META redirections.

```
public void detectMetaRedirectionsAndUpdate(boolean sleep) {
      // set the user-agent as Firefox for the client.
      final WebClient webClient = new WebClient(BrowserVersion.FIREFOX_3);
      // do not allow javaScript redirections to occur at this point.
      // not all JS redirections would be detected with htmlunit-rhino
      webClient.setJavaScriptEnabled(false);
      // do not allow redirections automatically.
      webClient.setRedirectEnabled(false);
      try {
            // get the final URL - which is the one we have after all ServerSide
            // redirections
            final HtmlPage page = (HtmlPage) webClient.getPage(finalUrl);
            Thread metaThread = new Thread(new Runnable() {
             public void run() {
                  // Check for META redirection
                  NodeList metas = page.getElementsByTagName("meta");
                  for (int i = 0; i < metas.getLength(); i++) {
                     HtmlMeta meta = (HtmlMeta) metas.item(i);
                     String httpEquiv = meta.getAttribute("http-equiv");
                     // If meta tag has http-equiv as refresh....
                     if (httpEquiv.equalsIgnoreCase("refresh")) {
                       String content = meta.getContentAttribute()
                            .replace(" ", "");
                       int start = content.toLowerCase().indexOf("url=") + 4;
                       String url = content.substring(start);
                       // if URL is the same as the original url => no meta
                       if (url.equals(finalUrl))
                         continue;
                       // else META redirection detected
                       cu.setHasMetaRedirection(true);
                       cu.RedirectionUrl.add(getComplete(url));
                       cu.setChainLength(cu.getChainLength() + 1);// increment
                  }
             }
            }
            });
            metaThread.start();
            metaThread.join();
       } catch (Exception e) {}
}
```

Figure A.2: Java code to detect META redirections using HtmlUnit

A.3 Detecting JavaScript redirections

Detection of JavaScript redirections requires a client with a DOM-aware JavaScript engine. Rhino (http://www.mozilla.org/rhino) is a JavaScript engine that can be used to detect redirections but it does not perform well for complex scripts due to the unavailability of a complete HTML DOM. User-Scripts are JavaScript code that can be injected into every page using tools such as GreaseMonkey (http://www.greasespot.net). Detection of redirections using user-scripts is a viable option but it does not scale well. Selenium (http://www.seleniumhq.org) is a test tool for web applications whose tests runs directly in a browser and is thus a good option to detect JavaScript redirections but was found to be complex to detect multiple redirections. Web browser controls such as SWT Browser Widget can be integrated inside Java applications that requries a complete browser. It provides interfaces to the browser to capture the browser events, the page events as well as inject script into the browser if required. It uses Mozilla, IE or Safari based on the operating system the application is currently running. We used SWT Browser widget for the interfaces it provides and also that it can act like a browser without actually displaying the browser windows which saves a lot of time on large-scale detection. Figure A.3 shows code segments that shows usage of SWT Browser widget and detection of redirections. We monitor the page unload event for the browser and a redirection is said to occur if there was no META redirection that was detected earlier. This way, we avoid the need for parsing JavaScript and at the sametime it also captures redirections that are invoked from rich components such as Flash or Silverlight.

```
//Add listeners on the browser (SwtBrowser)
private void addListeners() {
      // monitor the progress of the browser
      browser.addProgressListener(new MyBrowserProgressListener(this));
      // monitor the location changes in the browser.
      browser.addLocationListener(new MyBrowserLocationListener(this));
}
//Monitors the location changes in the browser
public void changing(LocationEvent arg0) {
    if (browser.getState() == BrowserState.LoadCompleted)
      browser.setState(BrowserState.Redirecting, arg0.location);
    if (browser.getState() != BrowserState.Navigating)
      browser.setState(BrowserState.Navigating, arg0.location);
  }
}
//Monitors the progress change of the browser
public void changed(ProgressEvent arg0) {
    // Browser Progress Changed Event
    if (arg0.current == 0 && arg0.current == arg0.total)
       browser.setState(BrowserState.LoadCompleted, browser.getUrl());
}
//Changes the state of the browser,detects redirection.
public synchronized void setState(BrowserState state, String url) {
    this.state = state;
    //If browser is redirecting and no META is detected
    if (this.state == BrowserState.Redirecting
        && !this.current.getHasMetaRedirection()) {
      //JS Redirection detected
      this.current.setHasJavaScriptRedirection(true);
      this.current.RedirectionUrl.add(url);
      this.current.setChainLength(this.current.getChainLength() + 1);
    }
}
```

Figure A.3: Java code to detect JavaScript redirections using SWT Browser Widget

Appendix B

Classification of Redirections - Heuristics and Techniques

In this chapter, we explain the heuristics that has been used to classify the redirections detected based on the reasons for redirections. Table B.1 displays a list of heuristics applied during the classification of redirections based on some of the reasons. These heuristics rely strongly on the naming conventions of the URL. Large websites,for maintainability, often tend to name their URL based on certain conventions which also appears user-friendly. The filters listed has been identified manually by processing several URL incrementally. As more filters were identified, the percentage of unclassified URL reduced and the process was stopped after having upto 80% of classification. Filter keywords such as r.msn, rd.yahoo are search in the hostname for the URL since those URL were observed to redirect for advertisments and these keywords were exclusive to classify URL crawled from yahoo.com and msn.com.

Not all reasons can be identified using reasons-heuristics. Reasons such as load balancing, virutal hosting do not rely on URL conventions based heuristics and instead are processed differently to achieve better classification results. For virtual hosting, source and destination URL are matched for hostnames and path. If the match count is high with destination URL having a different path, then we classify the particular redirection under virtual hosting. Classification for load balancing uses a similar technique except that we see if the redirection was internal and only match the hostnames. For example, redirection from URL with host http://www.shopping.com to a URL with host http://www0.shopping.com is classified under load balancing.

Table B.1: Heuristics for reasons-based classification

Reason	Filters
Security	login,auth,https,passport
Advertisements	ad,clic,buy,adtmt,overture,r.msn,rd.yahoo,doc.go
Information	article,data,r/b,feeds,feed,picks,doc,press
Indirection	dir,netring, directory,listing
Products	shop,g.msn,prod,games
Malware	tryfree, atdmt
Redirection Services	shorturl,tinyurl,url,tiny (domain-matches)
Spam Combat	msplinks.com
Localization	en-,/en/

The order in which the classifier applies the heuristics is also important. For example, a login page might have a redirection from a page with URL that satisfies Products (shopping pages) filter thereby misclassifies the redirection as products instead of security. Thus, it is required that the heuristics are applied in a particular order and the ordering has been improved after manual testing.

An important aspect of these heuristics is that the filters has been identified manually which makes the classification accuracy be different with other datasets. It was identified that redirections in the spam dataset cannot be classified with the reasons-heuristics described earlier since spam URL tend to use naming conventions to trick the users into clicking the links. A classification of spam redirections for reasons like cloaking, door way pages would be a worthy extension for the current classifier.

Appendix C

500 keywords used for the Keywords dataset

redtube
youtube
youporn
red tube
funny videos
maps
white pages
hotmail
free e-cards
craigs list
yahoo mail
sarah palin
cnn
home depot
wikipedia
kim kardashian
brazzers
free fta satellite keys
google.com
carmen electra
test
xhamster
scorpio
kristen archives
pisces
amazon
sagittarius
search engines
aquarius
free galleries
florida a
msn.com
aol
games
myspace
yahoo
facebook
yahoo.com
myspace.com
gambar pemerkosaan
redtube.com
youpron
game cheats
horoscope
hotmail.com
walmart
msn
coroas transando
gmail
dictionary
free lady sonia
aol.com
miley cyrus
lowes
cancer
sears
virgo
bikini
aries
leo
taurus
money
gemini
capricorn
ebay.com
cars
circuit city
pthc
google
ebay
craigslist
mapquest
you tube
youporn.com
xtube
tube8
music downloads
pedoland bbs pthc
bellsouthbiz.zip2.com
how high
girls
weather
pornotube
spankwire
amazon.com
best buy
target
__bfc_off george bush
youtube.com
nudist
fox news
hot
free ringback tones
libra
porntube
robb's celebrity page
fotos de bucetas
tube8.com
ikia furniture store
megarotic
dogpile
angelina jolie

paris hilton
map quest
lingerie
halloween costumes
google maps
tamil kama kathaigal
comcast.net
xtube.com
googletestad
espn
southwest airlines
youjizz
yellow pages
msnbc
www.yahoo.com
google earth
scary pumpkin stencils
plants
elweb bbs
drudge report
skinny dipping
megan fox
wedding flowers
girl
rihanna
flower guide
jennifer aniston
pink
black friday ads
al4a
barack obama
ask
katy perry
thong
family naturism
raven riley
we live together
tila tequila
brooke burke
lil kim exposed
cheapest
bbc news
free itunes codes
jab farm lessons
webkinz.com

jessica alba
boysfood
my space
maxporn
shufuni
mujeres nalgonas
lil wayne
jenna jameson
dogs
ls magazine
anime
facebook.com
roses
lindsay lohan
britney spears
webkinz
www.myspace.com
butterflies
music
dog
wicked weasel
bank of america
cleavage
types of flowers
videos
naruto
vanessa hudgens
breaking news
order pizza
naughty america
mapquest.com
limewire
walmart.com
funny
jonas brothers
naturist
driving directions
free games
kelly blue book
women
badjojo
imdb
nudists
bed bath and beyond
free online games

cheat planet
www.ls-magazine.com
www.zoo.com
mujeres eyaculando
air
jobs
jessica simpson
hearts
obama
new
yuvutu
pinay scandal
gmail.com
craigslist.com
weather.com
ask.com
cnn.com
news
myspace layouts
babes
pamela anderson
mature
tamil blue film
costco
people search
meaning of flowers
tube 8
board3.cgiworld.cgiid=j
doctor's excuse
www.youporn.com
one night in paris
utube
mp3
mother seduces son
fotos homens transando
madonna
asian
female photography
verizon
free
staples
olympics
spankwire.com
chochas de nenas
ups

travelocity
www.gadis bugil
halimbawa ng tula
paypal
reality kings
kid rock
jennifer hudson
kmart
micro bikini
runescape
pumpkin face patterns
hurricane ike
eskimotube
panties
asstr
emma watson fakes
emma watson
netflix
palin
dans movies
firefox
trish stratus
love
free desi stories
wii
daily horoscope
skimpy thongs
machine
traffic accidents
orbitz
auto trader
slipknot
okeanis.info
search engine
verizon wireless
tv guide
beach
imagefap
online games
food network
flickr
arcade.cc
satellite view of home
noaa
kwento ng ibong adarna

pornotube.com
usps
sublime directory
www.2spendless.com
free people search
movies
expedia
x tube
lyrics
airline tickets
photobucket
breasts
booty
contacts
aol mail
free movies
zip codes
wwe
www.hotmail.com
dump.com
tour de france
chris brown
craig's list
hannah montana
kohls
pic hunter
lolicon
robb's unofficial celeb
toni's spoiler page
itunes
boysfood.com
love horoscopes
coldplay
halle berry
printable sudoku puzzle
plentyoffish
topless
john mccain
law
qvc
blinkx
printable certificates
paypal.com
high school musical 3
toys r us

reverse lookup
song lyrics
www.redtube.com
bebo
puberty
office depot
internet
property
input
metacafe
thongs
vida guerra
kids
brazzers.com
metallica
talambuhay ni jose rizal
letras goticas
united airlines
halloween
naughty
american airlines
www.google.com
free als scan model
radio shack
ashley tisdale
drawing pumpkin faces
travel
christian
loli
free pumpkin cutouts
maikling talumpati
twilight
forced sissy stories
www.elkware.com
ikea
wild girls
ebony
used cars
cool pumpkin carvings
lucy pinder
tattoos
area codes
match.com
target.com
family guy

busty
sams club
male forced milking
baby names
printable pumpkin faces
people
anwar alaska
jessica biel
ls models
youjizz.com
beyonce
blackberry free themes
oops
malayalam blue films
cameron diaz
chinese horoscope
parts
shop
talambuhay ng mga bayani
adrienne bailon
g string
police radio frequencies
weather channel
chelsea football club
cuckold
dell
mga tulang pilipino
nfl
store
free music downloads
legs spread wide open
amtrak
vipzax
www.facebook.com
football
tamil stories
earth maps satellite
realtor.com
taylor swift
the rack
luis miguel desnudo
free music
free tattoo designs
talumpati ni rizal
drudge
soccer
pokemon
printable coloring pages
altavista
pogo
webmd
richards realm
printable bubble letters
recipes
cats
at&t
web md
family nudism
cabelas
funny pumpkin designs
free celebrity fake list
jokes
fireworks shows
black
amateur
linkin park
craiglist
printable sheet music
texas a
discount
desks
overstock.com
filter
latina
www.youtube.com
young
twin sisters kissing
web
pumpkin faces
free hermaphrodite
art
carribean
niagara falls
manga
ugly people
abba
met art
food
webkinz secret code
url
asian.vipzax.com
flowers
audio
rate my wife
dancing with the stars
ecards
maikling kwento
pet adoption
painted pumpkin ideas
round and brown
blonde
tera patrick
plenty of fish
undressing saree
pretty girls
obituaries
listen to music
play driving games
comcast
personal injury
mortgage calculator
avg
panochas gratis
world map
ink
free ringtones
denise masino
free exotic stories
webkinz ganz
sports
scary pumpkin patterns
michael phelps
breast
pumpkin stencils
grayvee
panochas mojadas
yahoomail
zodiac