Fraudulent and Malicious Sites on the Web

Ahmed Obied Reda Alhajj

Department of Computer Science

University of Calgary

Calgary, Alberta, Canada

{amaobied,alhajj}@ucalgary.ca

Abstract

Fraudulent and malicious web sites pose a significant threat to desktop security, integrity, and privacy. This paper examines the threat from different perspectives. We harvested URLs linking to web sites from different sources and corpora, and conducted a study to examine these URLs in-depth. For each URL, we extract its domain name, determine its frequency, IP address and geographic location, and check if the web site is accessible. Using 3 search engines (Google, Yahoo!, and Windows Live), we check if the domain name appears in the search results; and using McAfee SiteAdvisor, we determine the domain name's safety rating. Our study shows that users can encounter URLs pointing to fraudulent and malicious web sites not only in spam and phishing messages but in legitimate email messages and the top search results returned by search engines. To provide better countermeasures against these threats, we present a proxy-based approach to dynamically block access to fraudulent and malicious web sites based on the safety ratings set by McAfee SiteAdvisor.

Keywords: Fraudulent, Malicious, Web, Phishing, Pharming, Malware, Spam, Search Engines, Crawlers, Proxy

1 Introduction

The use of the web has grown in our daily lives. We use it to read news, watch movies, listen to music, communicate with friends, and research. In the last few years, the web has grown to an enormous size. Millions of domain names are being registered every day with no restrictions. This caused the number of warnings being made about the dark side of such technological revolution to increase and we are becoming vulnerable to many mysterious and malicious threats.

Attacks such as phishing and pharming pose significant threats to users' privacy. Phishing is a type of semantic attack in which victims are sent emails that deceive them into providing personal information and financial account credentials. Victims are directed to fraudulent web sites that are designed to trick them into divulging information such as credit card numbers, usernames, passwords, addresses, personal identification numbers (PINs), social security numbers and any further information that can be made to seem plausible. By hijacking brand names of banks, e-retailers, and credit card companies, phishers often convince the victims to respond [7]. The information phishers receive from the victims is then used to impersonate them as to empty their bank account, run fraudulent auctions, launder money, apply for credit cards, take out loans in their name, and so on [20]. According to the Symantec Internet Security Report [12], the Symantec Brightmail Anti-Spam system blocked 1.3 billion phishing emails in the first half of 2006.

A closely related attack to phishing is pharming. Pharming is an attack that manipulates the ways in which a user locates and connects to an organization's named hosts or services via DNS hijacking or poisoning. Users who navigate manually to a web site such as `www.bank.com` will believe they are visiting their bank's web site, while the pharmer stealthily usurps all web traffic directed at the victim domain [22]. Pharming attacks are harder to detect than ordinary phishing attacks since there is no need to lure victims to a fraudulent web site by sending emails when they will find the web site on their own [22].

Every year, billions of dollars are lost due to unsuspecting users entering personal and financial data into fraudulent web sites.

Another threat users can encounter on the web is malware (malicious software). Computer viruses, worms, Trojan horses, bots, spyware, and adware collectively termed malware, have become a significant threat to desktop security and integrity. Although malware can typically infect computers via a variety of ways, in this paper we focus on malware that installs itself surreptitiously by piggy-backing its code to legitimate software or by performing a drive-by download attack, exploiting vulnerabilities in the user's web browser. By infecting computers, malware authors can steal users' information and identities or use the infected computers to mount sophisticated attacks such as Distributed Denial of Service attacks, infect other computers, send spam, operate anonymously, or host questionable content on the compromised computers.

Companies encourage users to use anti-phishing tools and anti-virus protection software to detect fraudulent web sites and protect their computers from malware. A number of studies [19, 26, 18, 25, 23, 21], however, show how such protection mechanisms are ineffective. This is due to the fact that many users do not understand phishing and pharming attacks or how malware can infect their computers.

We conducted a study sampling web sites from different sources and corpora to show the density of fraudulent and malicious sites on the web. We extracted URLs from a legitimate email, spam, and phishing corpora. We used Yahoo!'s web search API to search for the top 10 searches in 2006 on Google and AOL, and a randomly selected 20 unedited real-time searches from Metaspy. For each query, we extracted the top 100-120 unique domain names appearing in the search results. Finally, we used two malicious web sites as seeds for our web crawler and crawled the web sites extracting URLs up to a specific threshold. For each URL, we extract its domain name, and use a local DNS server to resolve its IP address. We determine the domain name's geographic location using a database that maps IP addresses to geographic locations [6], and check if the web site is accessible by connecting to it and

checking the HTTP status code returned by the site's web server. We use Google [5], Yahoo! [15] and Windows Live [14] to check if the domain name appears in the search results, and McAfee SiteAdvisor [8] to get the site's safety rating. In our study, we encountered fraudulent and malicious web sites in every dataset we processed. We found that search engines do not filter out fraudulent and malicious web sites from their search results, and some of these web sites have high ranking scores.

To provide better countermeasures against fraudulent and malicious web sites, we present a proxy-based approach that relies on the safety ratings set by McAfee SiteAdvisor to prevent access to questionable web sites. We extended the source code for an open source Linux-based proxy server called Squid [11] and added features to check the site's safety rating before allowing HTTP requests to be forwarded.

The rest of the paper is organized as follows. Section 2 describes related work that places our current study and methodology in context. Section 3 and 4 describe data collection and analysis, and the results we obtained respectively. Section 5 describes our proxy filtering method. Finally Section 6 concludes and discusses the limitations of our study and methodology, and provides ideas for future work.

2 Related Work

Dhamija *et al.* [18] analyzed a large set of captured phishing attacks and developed a set of hypotheses about why these strategies work. To assess these hypotheses, a usability study was conducted in which 22 participants were shown 20 web sites and asked to determine which ones were fraudulent. 23% of the participants did not look at browser-based cues such as the address bar, status bar and the security indicators, leading to incorrect choices 40% of the time.

Jagatic *et al.* [19] conducted a study at Indiana University and showed how exploiting the social context can make phishing attacks far more effective. The researchers harvested freely available acquaintance data by crawling social network web sites such as MySpace and Facebook, and built a database with tens of thousands of relationships. In their study, the researchers focused on a subset of targets affiliated with

Indiana University by cross-correlating the data with the Indiana University's address book database. After building the database and relationships between students, the researchers sent phishing emails to University students aged 18-24 years old claiming to be from a friend. The phishing email led to a fraudulent web site that asked the students to provide their University's username and password. 72% of students provided valid usernames and passwords to the fraudulent web site.

Wu *et al.* [25] conducted two user studies of three security toolbars (SpoofStick, Netcraft, and SpoofGuard) and other security indicators such as the browser address and status bars to test their effectiveness at preventing phishing attacks. The researchers found that users fail to continuously check the browser's security indicators, since maintaining security is not the user's primary goal. Furthermore, the researchers found that users had no idea how sophisticated phishing attacks could be, and do not know good practises for staying safe online.

Zhang *et al.* [26] tested the effectiveness of 10 popular anti-phishing tools by using 200 verified phishing URLs and 516 legitimate URLs. The researchers found that only one tool (SpoofGuard) was able to consistently identify more than 90% of phishing URLs correctly; however, it also incorrectly identified 42% of legitimate URLs as fraudulent. The performance of other tools varied considerably depending on the source of the phishing URLs. Half the tools the researchers tested could correctly identify less than half of the phishing sites and many of these tools are vulnerable to some simple exploits.

Clayton [17] analyzed the current authentication protocols employed by online banking systems and found them to be entirely ineffective. Clayton proposed that simple changes in the authentication protocols can make the phisher's task significantly harder. These changes will require phishers to run real-time man-in-the-middle attacks and force them to persuade victims to perform unnecessary sensitive operations. Although being in "the middle" and dynamically altering the traffic is conceptually simple, there are a number of things that banks could do to ensure that it is far from straightforward. However, from the banks' point of view it may not be necessary to provide a perfect solution as long as the current

authentication protocols are "secure enough".

Moore *et al.* [20] analyzed empirical data on actual phishing web site removal times and user-response rates to better understand the impact of the take-down strategies being employed by phishing targets. In their analysis, the researchers found that sophisticated phishers (e.g., the rock-phish gang) are able extend the average lifetime of their phishing web sites. The researchers concluded that web site removal is part of the answer to phishing, but it is not fast enough to completely mitigate the problem.

The AOL/NCSA online safety study conducted a poll of 354 households, and also examined their computers for the presence of malware and examined their emails for the presence of phishing attacks [2]. The study showed that 23% of the respondents have received at least one phishing attempt via email. 56% of the computers examined were lacking virus protection, 44% were lacking a properly-configured firewall, and 38% were lacking spyware protection software. 12% of the computers were infected with at least 1 virus and 61% were infected with spyware.

The AOL/NCSA study did not attempt to identify how these computers became infected with viruses or spyware. It is very likely, however, that these computers were infected by visiting malicious web sites and/or downloading software from malicious web sites.

Researchers, e.g., [23, 21], and commercial companies such as McAfee SiteAdvisor [8] and Webroot [13] started using honeyclients to find threats on the web. Honeyclients represent one of the newest implementations derived from the idea of honeypots. In traditional honeypots, you set up a honeypot and wait for it to be probed, attacked, or compromised. A honeyclient, on the other hand, actively crawls the web seeking web sites that host malicious code or try to exploit vulnerabilities in client-side applications (e.g., web browser). Honeyclients mimic, either manually or automatically, the normal series of steps a regular user would make when visiting various web sites.

Ming *et al.* [23] developed an automated web patrol system called the Strider HoneyMonkey Exploit Detection System to automatically identify and monitor malicious web sites. The Strider HoneyMonkey

Exploit Detection System consists of a pipeline of monkey programs running possibly vulnerable web browsers on virtual machines with different patch levels and patrolling the web to seek out and classify web sites that exploit browser vulnerabilities. Within the first month of utilizing Strider HoneyMonkeys, 752 unique URLs hosted on 288 domain names attempted to exploit unpatched Windows XP machines when the monkeys crawled the URLs. One out of the 288 domain names was operating behind 25 exploit-URLs and was performing a 0-day [1] exploit of the javaprxy.dll vulnerability.

Moshchuk *et al.* [21] performed a large-scale, longitudinal study of the web, sampling both executables and conventional web sites using a web crawler for malicious objects. The study quantifies the density of spyware, the types of threats, and the most dangerous web zones in which spyware is likely to be encountered. In a May 2005 crawl of 18 million URLs, the researchers found spyware in 13.4% of the 21,200 executables they identified, and scripted "drive-by download" attacks in 5.9% of the web sites they processed.

Bragin [16] showed that spam messages are capable of carrying links to disconnected portions of the web. Bragin performed a study of spam using three sources: a spam honeypot, a group of high-spam student inboxes, and a newsgroup devoted to posting spam messages. Bragin found that 96% of URLs in spam messages point to web sites not reachable by crawlers and that most of these sites are not reviewed for safety by security companies.

3 Data Collection and Analysis

The approach we used in our study can be divided into two parts. First, we collected data from different sources and corpora. Second, we extracted URLs from the collected data and tried to gather more information about them. In this section we describe the two different parts in more detail.

[1] In this paper, a 0-day exploit refers to a vulnerability exploit that is released before, or on the same day the vulnerability is released to the public.

3.1 Data Collection

Our goal was to examine as many URLs as possible from a variety of sources and corpora in a relatively short period of time. We considered the following sources and corpora in our study:

3.1.1 Legitimate Email, Spam, and Phishing Corpora

We downloaded the 2006 TREC public spam corpora from [1] which contain 37,822 email messages: 12,910 legitimate messages and 24,912 spam messages. We processed the corpora to separate legitimate messages from spam messages, and drop messages encoded in base-64 since such encoding is used to encode attachments. We decided to ignore messages with attachments to avoid the overhead of decoding such messages. After processing the corpora, we ended up with 36,391 messages: 12,594 legitimate messages and 23,797 spam messages. We used a tool we developed to harvest URLs from a given corpus and extract their domain names. With the tool, we were able to extract 15,511 domain names from the legitimate email corpus and 40,863 domain names from the spam corpus. We found 2,229 unique domain names in the legitimate email corpus and 822 unique domain names in the spam corpus. For the phishing corpus, we downloaded 3 UNIX mbox files from [10] that have a total of 2,271 messages. We extracted 17,668 URLs from the phishing messages with 893 unique domain names.

3.1.2 Top AOL and Google searches in 2006

We looked at the top searches in 2006 on AOL [3] and Google [4] which can be found in table 1. We signed up for a Yahoo! email account and obtained an application ID to use Yahoo!'s web search API. We developed a tool that takes a query and uses the API to find the top 100-120 important and relevant URLs. For each query in table 1, we used the API to obtain the relevant URLs. When you search for a query using the API, Yahoo! returns the search results in an XML file. With the URL harvester tool we described earlier, we harvested the URLs from the returned XML files and extracted the unique domain

names. The total number of unique domain names we found for the top 2006 searches on AOL is 751, and the total number of unique domain names we found for the top 2006 searches on Google is 663.

Table 1. Top 10 searches in 2006 on AOL and Google

Rank	AOL	Google
1	Weather	Bebo
2	Dictionary	MySpace
3	Dogs	World Cup
4	American Idol	Metacafe
5	Maps	Radioblog
6	Cars	Wikipedia
7	Games	Video
8	Tattoo	Rebelde
9	Horoscopes	Mininova
10	Lyrics	Wiki

3.1.3 Metaspy searches

Metaspy [9] is a search spy web site that lists real-time web searches occurring on the Metacrawler search engine. On May 30th 2006, we scraped 20 random unedited searches from Metaspy and used Yahoo!'s web search API to find the top important and relevant URLs as described earlier. The total number of unique domain names we extracted is 1,402.

3.1.4 Web crawls

We developed a web crawler that takes a web site as a seed and traverses the web site in a breadth-first fashion to extract unique domain names up to a specific threshold. We used our web crawler to crawl two sites: `www.crack.ms` and `www.screensavers.com`. The first web site is a known malicious web site

located in Russia which appears as the 2nd URL when you search for the keyword "cracks" on Google. The second web site is also a known malicious web sites located in the United States which hosts software bundled with spyware and adware. The second web site appears as the 2nd sponsored URL on Google when you search for the keywords "free" and "wallpapers".

The threshold (upper bound) we used in our web crawler is 2,300. The web spider crawled the two web sites and extracted 2,181 unique domain names from `www.crack.ms` and 2,248 unique domain names from `www.screensavers.com`.

Table 2. Data Collection Summary

Source	URLs	Domains
Legitimate Email Corpus	15,511	2,229
Spam Corpus	40,863	822
Phishing Corpus	17,668	893
AOL searches	1,189	751
Google searches	1,143	663
Metaspy searches	2,353	1,402
www.crack.ms crawl	2,181	2,181
www.screensavers.com crawl	2,248	2,248
Total	83,156	11,189

3.2 Data Analysis

The total number of URLs we found is 83,156 hosted on 11,189 unique domain names. We argue that the domain name collection we analyze is representative for the purpose of our study. We looked at a large number of URLs from a variety of sources and corpora, and we gathered more information about the URLs from different sources. We developed a domain name analyzer tool to perform the information gathering process. We used a hash table to store the domain names we process where the names are

used as keys. For each key, we maintain a node which stores the following:

- Frequency
- IP address
- HTTP status code
- Google index
- Yahoo! index
- Windows Live index
- McAfee SiteAdvisor rating
- Geographic Location

3.2.1 Frequency

The total number of occurrences (frequency) of a domain name in a given corpus is an integer. When a domain name is added to the hash table for the first time, the frequency value corresponding to that domain name is initialized to 1. Every time there is a collision in the hash table which implies encountering another occurrence of an existing domain name, we increment the frequency value by 1.

3.2.2 IP address

For each domain name we encounter, we use a local DNS server to resolve its IP address. If the DNS server cannot resolve the domain name or does not respond within 15 seconds then we set the IP address to "N/A".

3.2.3 HTTP status code

The HTTP status code is a string used to store the the status code returned by the web server which is pointed to by a given domain name. We open a connection and send an HTTP GET request to the web server and wait for the response. The timeout we used in our study is 15 seconds. If the web server does not respond within 15 seconds then we set the HTTP status code to "N/A". Otherwise, we check the response. If the response is a redirect (3xx status code) then we follow the redirect until we hit the final web server and store the response the final web server returns.

3.2.4 Google, Yahoo!, and Windows Live

We used 3 search engines to check if a domain name appears in the search results. The index for each search engine is a string which can be one of two: "indexed" to indicate that a domain name appears in the search results and "not indexed" to indicate that a domain name does not appear in the search results.

For each search engine, we first construct an HTTP GET request to include the given domain name as a query and then send it to the search engine's web server. To avoid exhausting the web servers by issuing too many requests, we inserted a 2 second delay between requests. Every time a response is received, we parse the returned web page searching for sentences indicating that the given domain name cannot be found and set the index string accordingly.

3.2.5 McAfee SiteAdvisor safety ratings

McAfee SiteAdvisor [8] use automated web spiders to crawl the web and test every web site, download, and email sign-up form. Web sites are tested for excessive pop-ups, fraudulent practises, and web browser exploits. Downloads are tested for viruses, and bundled adware, spyware, or other unwanted programs. Finally, sign-up forms are completed using a one-time use email address so any subsequent spam can be

tracked. For each visited web site, McAfee SiteAdvisor assigns a safety rating [8]:

- **Green (safe):** No significant problems were found.
- **Yellow (caution):** Minor security or nuisance issues were found. Also, applies to web sites that have previously had past security issues.
- **Red (warning):** Serious security issues were found (e.g., hosting malicious code, sending excessive spam messages, exploiting web browser vulnerabilities, etc).
- **Gray:** The web site has not been tested.

In our analysis, we rely heavily on McAfee SiteAdvisor to indicate whether a given web site is fraudulent or malicious. We contacted McAfee SiteAdvisor and obtained permission to use their data in our study. For every domain name, we construct an HTTP GET request with the right parameters and send it to McAfee SiteAdvisor's web server. We then parse the response, extract the safety rating, and set the safety rating string in our nodes accordingly. The rating can be one of: green, yellow, red, or gray. To avoid exhausting the web server with too many request, we inserted a 2 second delay between requests.

3.2.6 Geographic Location

Hostip.info [6] is a community-based project to geolocate IP addresses. The database maintained by hostip.info is freely available. We used it to map the IP addresses we have to their geographic locations. For each IP address, we construct HTTP GET requests to hostip.info's web server, parse the response, and set the geographic location string accordingly. If hostip.info cannot find the geographic location for a given IP address or it does not respond within 15 seconds then we set the geographic location to "N/A". Again we inserted a 2 second delay between requests to avoid exhausting the web server with too many requests.

Table 3. Data Analysis Summary

Source	Domains	Accessible	No Backlinks			McAfee SiteAdvisor		
			Google	Yahoo!	Windows Live	Red	Yellow	Gray
Legitimate Corpus	2,229	97.82%	0.26%	0.35%	0.98%	0.44%	1.3%	7.17%
Spam Corpus	822	91.66%	18.49%	18.97%	24.20%	12.65%	1.70%	12.65%
Phishing Corpus	893	90.69%	15.56%	11.75%	20.04%	2.01%	0.78%	35.49%
AOL searches	751	99.33%	0%	0%	0%	2.39%	2.13%	1.33%
Google searches	663	99.24%	0.15%	0%	0.90%	1.80%	1.80%	9.50%
Metaspy	1,402	99.06%	0.28%	0%	2.85%	3.85%	2.56%	9.70%
www.crack.ms	2,181	98.2%	0.73%	0.87%	19.71%	27%	2.65%	17.51%
www.screensavers.com	2,248	98.43%	0.22%	0.75%	1.29%	1.86%	1.60%	9.69%
Overall	**11,189**	**96.80%**	**4.46%**	**4.08%**	**8.74%**	**6.5%**	**1.81%**	**12.88%**

4 Results

4.1 Legitimate Email Corpus

From the 2,229 unique domain names we processed, 2,181 returned a 200 OK HTTP status code which implies they are accessible. The domain name with the highest frequency (appeared 1,204 times) is `www.pdfzone.com` which is located in Canada. 6 domain names were not found on Google, 8 were not found on Yahoo!, and 22 were not found on Windows Live. 2,029 domain names have a green verdict, 30 have a yellow verdict, 10 have a red verdict, and 160 have a gray verdict.

8 of the red domain names are located in the United States, 1 is located in China, and 1 is located in Russia. The red domain name with the highest frequency (appeared 9 times) is `www.mail.com`. Most of the red domain names were found to be distributing software bundled with spyware and adware.

4.2 Spam Corpus

Although the spam corpus had the highest number of URLs (40,863) in it, we only found 822 unique domain names. This gives us an indication of the number of domain names involved in flooding the world with spam. 758 out of the 822 domain names returned a 200 OK HTTP status code. The domain name with the highest frequency (appeared 2,821) is `g-images.amazon.com` which belongs to amazon.com. This is due to the fact that the spam corpus we processed seems to have phishing messages in it. Finding a domain name that belongs to a legitimate company with a high frequency indicates how this legitimate company seems to be the favourite web site to masquerade by phishers.

152 of the domain names we processed were not found on Google, 156 were not found on Yahoo!, and 199 were not found on Windows Live. 600 domain names have a green verdict, 104 have a red verdict, 14 have a yellow verdict, and 104 have a gray verdict. We believe there is a high number of green domain names in the corpus because of the phishing messages. The phishers are using URLs that point to resources (e.g., images) located at legitimate companies' domains such as amazon.com, ebay, and paypal.

The red domain name with the highest frequency (appeared 2,700 times) is `left.anotherparty.net` which is located in the United States. The second highest (appeared 1,656 times) is `huator.com` which is also located in the United States. The interesting part is that both domain names point to the same IP address: 204.160.156.44. We also found another 33 domain names pointing to 204.160.156.44. Out of the 104 red domain names, there are only 34 unique IP addresses. These IP addresses are in the USA, Canada, China, Germany, Russia, Denmark, and Australia.

4.3 Phishing Corpus

Out of the 893 domain names we found, 810 are still accessible. Most of the domain names are pointing to resources on legitimate companies' web servers such as amazon.com, ebay, paypal, etc. The domain name with the highest frequency (appeared 3,061 times) is `pics.ebaystatic.com` which belongs to ebay. The second highest (appeared 1,989 times) is `images.paypal.com` which belongs to paypal. 139 of the domain names were not found on Google, 105 were not found on Yahoo!, and 179 were not found on Windows Live. 550 domain names have a green verdict, 18 have a red verdict, 7 have a yellow verdict, and 317 have a gray verdict.

This corpus has the highest number of gray domain names (35.49%). We believe these gray domain names could have been assigned a red or a yellow verdict if they were tested by McAfee SiteAdvisor. The red domain names with the highest frequency (appeared 23 times each) are `mx1930.aa04.com`, `mx20211.aa03.com`, and `mx18198.ff02.com` which are all located in the United States, and returned a 403 Forbidden HTTP status code when we tried to access them. These domain names and other domain names using a similar naming convention are involved in heavy spamming operations.

4.4 AOL, Google, and Metaspy searches

For the top AOL searches, 746 out of the 751 domain names we processed are accessible. All the domain names we processed appear on Google, Yahoo!, and Windows Live. 707 domain names have a green verdict, 18 have a red verdict, 16 have a yellow verdict, and 10 have a gray verdict. The red domain names with the highest frequency (appeared twice each) are `www.armorgames.com`, `www.screensavers.com`, `www.swirve.com`, `games.excite.com`, `www.newgrounds.com` in the United States, and `homepages.pathfinder.gr` in Greece. The yellow domain name with the highest frequency (appeared twice) is `horoscopes.myway.com` in the USA.

For the top Google searches, 658 out of 663 domain names are accessible. 1 domain name was not found on Google and 6 were not found on Windows Live. 567 domain names have a green verdict, 12 have a red verdict, 12 have a yellow verdict, and 63 have a gray verdict. The red domain names with the highest frequency (appeared twice each) are `www.vivid.com`, `www.mininova.com`, `eng.anarchopedia.org.com` in the United States, and `www.freedownloads.be` in Belgium.

For the Metaspy searches, 1,389 out of the 1,402 domain names are accessible. 4 domain names were not found on Google and 40 were not found on Windows Live. 1,176 domain names have a green verdict, 54 have a red verdict, 36 have a yellow verdict, and 136 have a gray verdict. Most of the red domain names are located in the United States.

Most of the domain names we processed appear on Google, Yahoo!, and Windows Live. These domain names are somewhere in the top 100-120 search results which indicate they have a high ranking score given the enormous size of the web. Although one would expect search engines to filter out fraudulent and malicious web sites from their search results to avoid referring users to these web sites, our analysis shows the existence of these web sites in the top search results.

4.5 Web crawls

4.5.1 www.crack.ms

2,142 out of the 2,181 domain names we extracted by using `www.cracks.ms` as a seed in our web crawler returned a 200 OK HTTP status code. The crawl revealed the highest density of red domain names (27%) and yellow domain names (2.65%). 1,152 domain names have a green verdict, 589 have a red verdict, 58 have a yellow verdict, and 382 have a gray verdict.

We found 132 unique IP addresses for the 589 red domain names. 54 IP addresses are in the Netherlands, 40 are in the United States, and the rest are spread across Russia, United Kingdom, Hong Kong, Germany, Poland, Ukraine, Italy, Czech Republic, Iran, and Sweden. The domain names in the Nether-

lands have the following form: `software-crack-page-x.serialcrackz.com` where x is a letter followed by a number (e.g., n2, y1, t13, etc). These domain names were found to exploit browser vulnerabilities to install spyware and adware, and are involved in spamming and phishing operations.

4.5.2 www.screensavers.com

2,213 out of the 2,248 domain names we extracted from `www.screensavers.com` are accessible. 5 domain names were not found on Google, 17 were not found on Yahoo!, and 29 were not found on Windows Live. 1,952 domain names have a green verdict, 42 have a red verdict, 36 have a yellow verdict, and 218 have a gray verdict. Most of the red and yellow domain names are located in the United States.

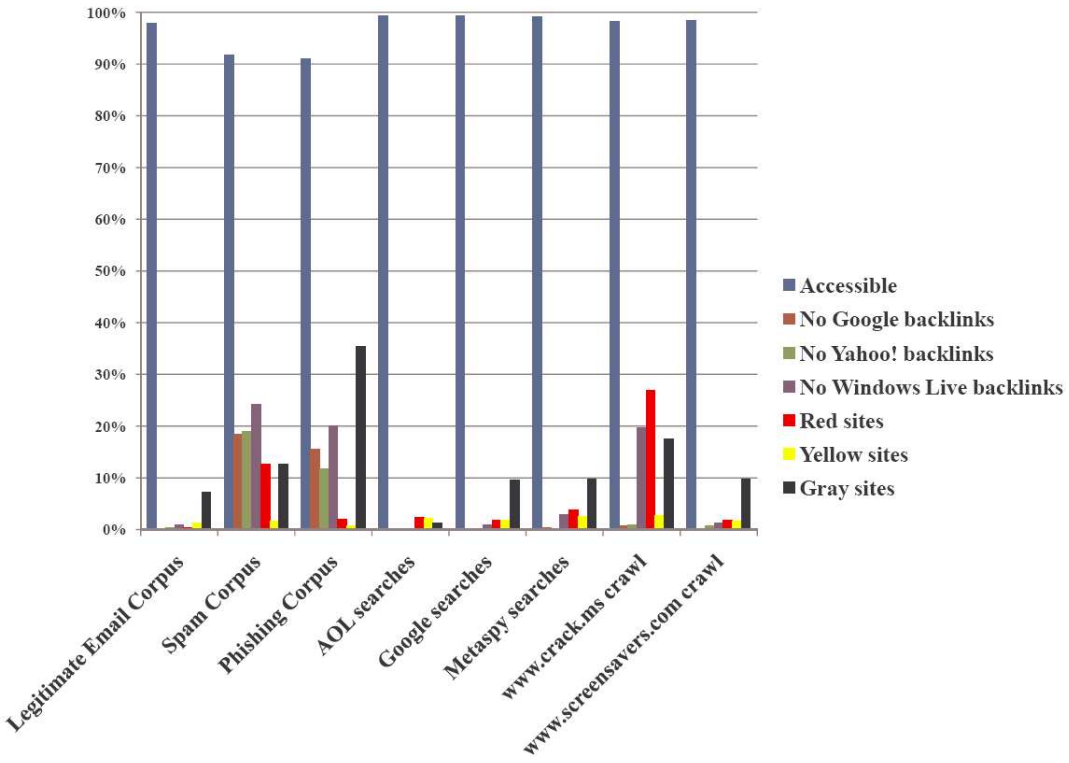

Figure 1. Analysis results for different web sites

5 Countermeasure

To provide better countermeasures against fraudulent and malicious web sites, we developed a proxy-based method to prevent access to such web sites dynamically and based on the safety ratings set by McAfee SiteAdvisor. Although McAfee SiteAdvisor provides a browser extension to alert users about questionable web sites, usability studies [18, 25, 26] have shown that browser extensions are ineffective especially when used by inexperienced users. In a dynamic network environment where users can log in and out of a network any time, it becomes challenging to control what web sites they can and cannot access. However, by using filtering software at the gateway of a network, one can enforce access rules.

Our method is an extension to an existing open source software called Squid. Squid [11] can act as both a proxy and a cache. However, in this paper we only focus on the proxy part of Squid. Squid accepts a request from a client, processes that request, and then forwards the request to a web server [24]. The request may be logged, rejected, and even modified before forwarding. Squid has two parts. The first part, which is called the client-side, talks to web clients (e.g., browsers and user-agents). The second part, which is called the server-side, talks to HTTP, FTP, and Gopher servers.

Squid has static built-in functionality to do access controls. You can specify the IP address, domain name, request method, or server port number to block by adding the appropriate rules to an access control list in Squid's configuration file. Since everything has to be specified before Squid starts, Squid has to be restarted if anything in the access control list changes. To overcome this issue and make Squid capable of performing dynamic access controls, we modified its source code and added our own implementation. Our implementation has three safety modes which can be set with a new flag we added to Squid. These modes are:

- **High:** Blocks access to red, yellow, and gray web sites.
- **Medium:** Blocks access to red and yellow web sites.

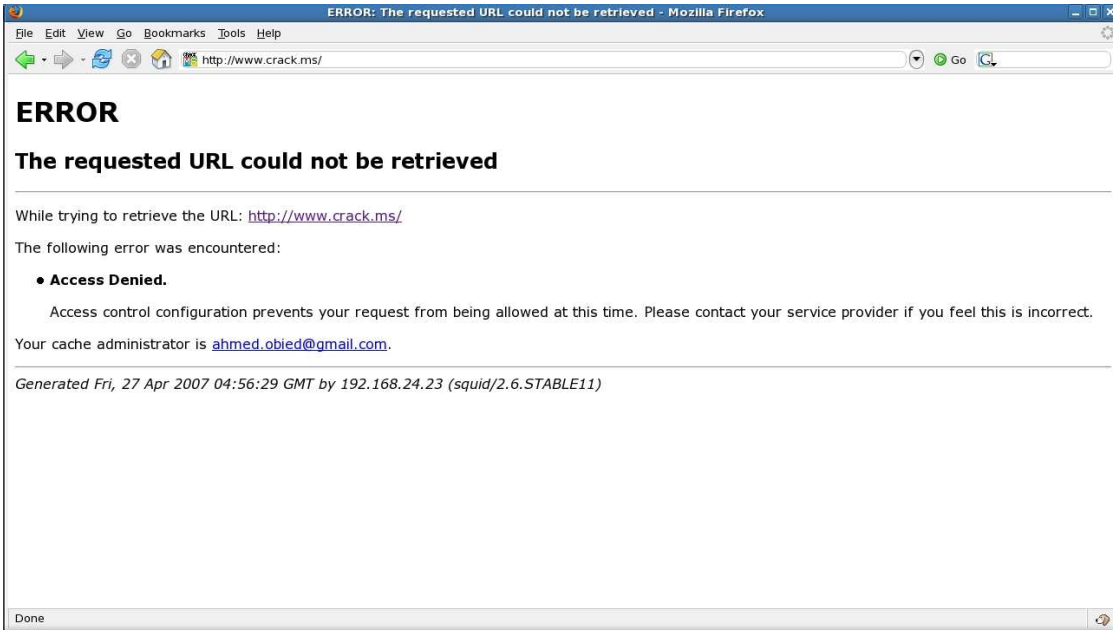

Figure 2. "Access Denied" redirect on Squid

- **Low:** Blocks access to red web sites.

Before Squid's server-side takes a URL and resolves its IP address to communicate with the web server, our implementation connects to McAfee SiteAdvisor's server and sends a GET request with the domain name of the URL used as a query. We parse the response and extract the domain name's safety rating: green, yellow, red, or gray. Based on the safety mode used, we tell Squid to connect to the web server and forward the traffic or send the client an "Access Denied" page.

We tested our methodology by using different safety modes and accessing some of the red, yellow, and gray domain names we encountered in our study. Although our method worked with minimum overhead, we believe it can be improved by using a local cache to store the safety ratings instead of connecting to McAfee SiteAdvisor's server every time a URL is given. Since our access to McAfee SiteAdvisor's database is limited, we decided to leave this strategy as future work where we hope to build and use our own safety ratings database.

6 Conclusion

The goal of this research is to quantify the density of fraudulent and malicious sites on the web and provide a better countermeasure strategy. To do this, we harvested URLs from different sources and corpora and conducted a study to examine these URLs in-depth. We harvested URLs from a legitimate email, spam, and phishing corpora. We examined the top URLs returned by Yahoo! for the top 10 searches on AOL and Google, and a 20 random unedited real-time searches from Metaspy. Finally, we examined URLs extracted from two different malicious web sites that appear in the top search results on Google. For each URL, we extracted its domain name. We used a local DNS server to resolve the domain name's IP address and mapped the IP address to a geographic location. We checked 3 search engines (Google, Yahoo!, and Windows Live) to find whether the domain name appears in the search results, and used McAfee SiteAdvisor to determine the domain name's safety rating.

Overall, our results show that the density of fraudulent and malicious sites on the web is substantial. Although it is difficult to generalize from a single study, but from the datasets we processed, fraudulent and malicious sites were found in every dataset. Overall, 6.5% of the domain names we processed have a red verdict, 1.81% have a yellow verdict, and 12.88% have a gray verdict. The domain names with gray verdicts are most probably fraudulent or malicious web sites. If these numbers are even close to representative web sites visited by users, it is not surprising that many user continue to be victims of fraud and malware.

Although we relied heavily on McAfee SiteAdvisor's safety ratings in our analysis and countermeasure strategy, we intend to work on building our own safety ratings database in the future. Relying on a commercial third party database has its limitations. We hope to learn from the study we conducted in providing a better site safety database by using more efficient and robust crawling strategies that cover a large portion of the web.

7 Acknowledgements

Thanks to McAfee Inc. for granting us permission to use SiteAdvisor's safety ratings in this research.

References

[1] 2006 TREC Public Spam Corpora. plg.uwaterloo.ca/~gvcormac/trecspamtrack06/.

[2] America Online and the National Cyber Security Alliance. AOL/NCSA online safety study. www.staysafeonline.info/pdf/safety_study_2005.pdf, December 2005.

[3] AOL hot searches in 2006. about-search.aol.com/hotsearches2006/index.html.

[4] Google hot searches in 2006. googlesystem.blogspot.com/2006/12/top-searches-on-googlecom-in-2006.html.

[5] Google. www.google.com.

[6] host.info. www.hostip.info.

[7] McAfee. Phishing and Pharming: Understanding phishing and pharming. www.mcafee.com/us/local_content/white_papers/wp_phishing_pharming.pdf, January 2006.

[8] McAfee SiteAdvisor. www.siteadvisor.com.

[9] Metaspy. www.metacrawler.com/info.metac/searchspy.

[10] Online Phishing Corpus. monkey.org/~jose/wiki/doku.php?id=PhishingCorpus.

[11] Squid. www.squid-cache.org.

[12] Symantec Corporation. The Symantec Internet Security Threat Report. www.symantec.com/enterprise/threatreport/index.jsp, September 2006.

[13] Webroot Software, Inc. Automated threat research. `research.spysweeper.com`.

[14] Windows Live. `www.live.com`.

[15] Yahoo!. `www.yahoo.com`.

[16] T. Bragin. Measurement Study of the Web Through a Spam Lens. Technical Report TR-2007-02-01, University of Washington, Computer Science and Engineering, 2007.

[17] R. Clayton. Insecure Real-World Authentication Protocols (or Why Phishing is so Profitable). *13th International Workshop on Security Protocols, Cambridge, UK*, 2005.

[18] R. Dhamija, J. Tygar, and M. Hearst. Why Phishing Works. In *Proceedings of the SIGCHI conference on Human Factors in Computer Systems*, 2006.

[19] T. Jagatic, N. Johnoson, M. Jakobsson, and F. Menczer. Social Phishing. In *Communications of the ACM. To appear*, 2007.

[20] T. Moore and R. Clayton. An Empirical Analysis of the Current State of Phishing Attack and Defence. *6th Workshop on the Economics of Information Security*, 2007.

[21] A. Moshchuk, T. Bargin, S. Gribble, and H. Levy. A Crawler-based Study of Spyware on the Web. In *Proceedings of the Internet Society Network and Distributed System Security Symposium (NDSS)*, 2006.

[22] S. Stamm, Z. Ramzan, and M. Jakobsson. Drive-By Pharming. Technical Report TR641, Indiana University, Department of Computer Science, 2006.

[23] Y. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. In *Proceedings of the 14th USENIX Security Symposium*, 2005.

[24] D. Wessels. *Squid: The Definitive Guide*. O'Reilly and Associates, 2004.

[25] M. Wu, R. Miller, and S. Garfinkel. Do Security Toolbars Actually Prevent Phishing Attacks. In *Proceedings of the SIGCHI conference on Human Factors in Computer Systems*, 2006.

[26] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding Phish: Evaluating Anti-Phishing Tools. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007)*, 2007.