Tracking Malware With Public Proxy Lists

GIAC (GSEC) Gold Certification

Author: J. Laird Powers, jaspowers@hotmail.com
Advisor: Richard Wanner

Accepted: TBD

Abstract

Public open proxy lists have been compiled and published on the Web for over a decade. A simple search for “proxy list” will return hundreds of thousands of results from sites offering “anonymity” and “privacy” for Web surfers, often as a come-on for paid, “Elite” services. The pages of many of these sites list hundreds of Internet Protocol (IP) addresses and port numbers of hosts across the world. Individually, these sites represent a nuisance, but collectively they contain a massive amount of data that can be leveraged to ascertain and often predict the spread of certain forms of malware. With simple tools, it is possible to establish a baseline of known proxies, monitor the most active sites, and track the spread of new proxies over time, often with surprising results. A two and a half year study of proxy lists demonstrates that evidence of the Koobface worm appeared in these lists months before press reports of its spread appeared. It is recommended that the security community monitor the valuable information these lists contain.

1. Introduction

1.1. A History of Proxies and Malware 1990-2010

The Web was born on Christmas Day, 1990 when the CERN Web server (CERN httpd 1.0) went online. By version 2.0, released in 1993, CERN httpd, was also capable of performing as an application gateway. By 1994, content caching was added. With the publication of RFC 1945 two years later, proxy capabilities were forever embedded into the HTTP specification (Berners-Lee, Fielding, & Frystyk, 1996).

With that blessing, proxies followed the explosive growth of the Internet, with proxy servers being built into or riding on top of firewalls, gateways, and Web servers. The main benefits of proxies—content caching and bandwidth saving—enabled many companies and institutions to leverage their limited resources to bring the Web in-house.

And in 1999, when most of the IT world was preoccupied with the Y2K "threat", a new virus with an interesting and previously unseen *modus operandi* appeared behind the scenes. Instead of infecting executable files or sending massive amounts of email, the RingZero Trojan contained no obvious malicious code. Instead, it silently scanned the Internet for open proxies, sending the results of these searches back to its master, a system at *www.rusftpsearch.net*, which was quickly taken down. (NIPC, 1999).

While malware authors were planting proxies, open Web proxies became a commodity, leveraging the *third* benefit of proxies most had considered a drawback: anonymity. By obfuscating his Internet Protocol (IP) address with the proxy's IP address, a proxy user can essentially disguise his location. This is by far the main driver for open proxy use by the non-malware user today, but there is a fourth, unintended, benefit: bypassing Web filtering.

As companies and campuses opened up to the Web a new problem surfaced. Employees and students were abusing Web services and violating Web Acceptable Use Policies (AUPs) daily. New products, often leveraging proxy services themselves, appeared on the scene to control these users' Web habits. Almost immediately these users retaliated by using external proxy servers to bust through firewalls and Internet filters.

And there were plenty to go around. Unsecured, open Web proxies were literally everywhere, in every major city with a fat pipe: universities, Internet cafes, businesses, hotels, hospitals, Internet Service Providers (ISPs), and telecoms. There were even proxies where there weren't supposed to be proxies (Prosise & Shah, 2001). Compaq, the server manufacturer, inadvertently coded a proxy server into their Insight Server Management Agents on TCP port 2301 (US-Cert, 2001), a port that is routinely scanned by proxy hunters to this day, nine years after it was patched.

Lists of open proxies became commonplace and a simple Web search could bring up hundreds of active proxies. Abuse was rampant during 2000-2005, when the naiveté of systems administrators and security-unconscious consultants was at its peak. At the same time, the security industry was beginning to thrive. Open proxies were about to take a hit, and their numbers drastically declined, beginning mid-decade.

Meanwhile, the malware "industry" was maturing as well.

Sometime after RingZero, as malware authors became more professional they came to the realization that it is more expedient to *install* a proxy on a vulnerable system than it is to install a proxy *scanner* on a vulnerable system. The benefits are immediately realized. Proxy functionality would become part of the standard toolset for Trojans and botnet agents in the decade following RingZero, to the point that one would be hard-pressed to find modern malware *without* proxy services built in. Memorable malware with proxy tricks includes MyDoom, Bagel, Sality, Koobface, and a host of smaller players like Xorpix, Fackemo, Ranky, Proxy-DD, Cidra, and Daemonize. Malware proxies are leveraged for distributed denial of service (DDoS) attacks and click fraud. Malware proxies grew as legitimate proxies were locked down. Proxy list compilers scooped them up and published or sold them.

Along with the increased awareness of IT security issues in the workplace came a locking down of the ports accessible to the average business or institutional user. The standard proxy ports (8080, 3128, etc.) were commonly blocked. The "legitmate" proxy listers, whose typical proxy market segment was the user seeking "privacy" and avoiding "censorship" turned instead to listing PHP and CGI proxies (hereafter referred to as "Web proxies"), available over standard, unblocked ports (80 & 443) over standard HTTP URLs. Leveraging advertising programs such as Google Adsense, AdBrite, and others allowed them to monetize their sites without explicitly charging for the service. Some offer their own advertising services, pushing unwanted popup and pop-under ads to their users. Most offer proxy lists because the base URLs change constantly as they attempt to evade Web blocking software, a practice that gets more difficult as active Web scanning appliances become more prevalent.

Today, the Web proxy lists dominate search results. Proxy.org is the main site popularizing these services to both end-users and Web proxy admin wannabees. The owner, one "Baron Munchuasen", has even acquired the most popular Open Source PHP proxy script, glype (Munchausen, 2010).

ProxyFire is the most active site catering to the "old school" scan-and-collect crowd, offering their own proprietary tools for sale, and offers a good overview of the ways open proxies are leveraged by the average "gray hat" user.

Although Web proxies are an interesting study in themselves, open proxies offer a far better view of the spread of malware. As long as malware and botnets flourish there will be open proxy lists. Collection and analysis of these lists should be practiced at the highest levels of national security for insights into the movements and tactics of botnets, the foot soldiers of future cyberwars (Mannes & Hendler, 2009).

1.2. Notable Examples of Proxy Abuse

1.2.1. CoDeeN

CoDeeN is a Content Distribution Network (CDN) based on open CERN proxies (primarily SQUID) that went live in 2003. By their own words, the decision to go live with open proxies defied "conventional wisdom and standard practices" (Pai, Wang, Park, Pang, & Petersen, 2004). Opting for simplicity and assuming the unpublicized nature of the project would not attract unwanted users; they soon learned the error of their ways. Within days of the network becoming stable, reports of abuse began to come in.

The administrators soon discovered that CoDeeN proxies were listed on various sites promoting proxies and ancillary software designed to test and leverage open proxies for bulk emailing. They suddenly found themselves with a new project: mitigating the abuse of their network. And, not surprisingly for the time, there was little published data on how to go about doing this. They found they were pioneers in a field they didn't know existed. But by limiting daily bandwidth, allowing only HTTP GET and HEAD requests from "outside" users, and greeting users with a stern warning about CoDeeN's willingness to work with local authorities in abuse reports, the problem was eventually brought under control (Pai, Wang, Park, Pang, & Petersen, 2004).

The CoDeeN operators identified five basic categories of abuse (Pai, Wang, Park, Pang, & Petersen, 2004), all of which are still leveraged today by proxy (ab-) users.

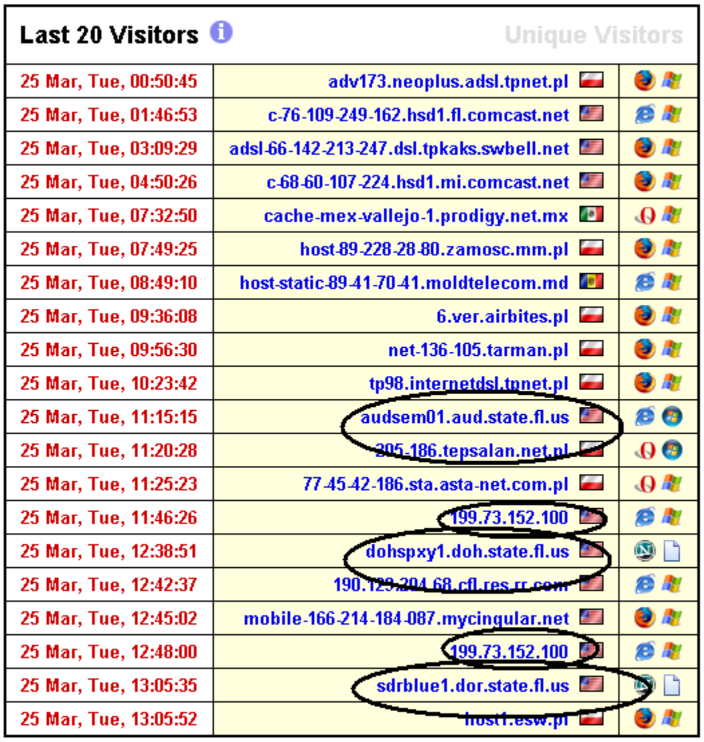
- *Spammers* — Early proxy servers offered the ability to tunnel TCP application ports using the "CONNECT" directive, but this functionality, heavily abused during the 1990s, was limited to SSL connections by the time CoDeeN went online. Nevertheless, they received thousands of CONNECT requests for SMTP (port 25) per day. SPAM was also propagated through CoDeeN by malicious use of Web mail and IRC (Internet Relay Chat).
- *Content Theft* — This issue was specific to the University environment CoDeeN operated in and their external providers' choice of IP address authentication, which allowed any user of the proxy network to be authenticated by virtue of using the network itself.
- *High Request Rates* — Typified by brute-force password attacks, Google crawlers, and click fraud.
- *Bandwidth Hogs* — The network was heavily saturated by bulk data transfers, leveraging the high bandwidth typically available in Universities.
- *Anonymity* — The CoDeeN operators discovered their addresses published in anonymous proxy lists. Since they did not, at the time, prohibit the CONNECT method on port 80 (HTTP) they found they were hosting a variety of private TCP tunnels.

Even though the CoDeeN network is not as wide open as it used to be, CoDeeN proxies can still found on most proxy lists published on the Web. However, most proxy list aficionados consider them "filler" and they are generally disliked intensely because of the limitations.

1.2.2. State of Florida

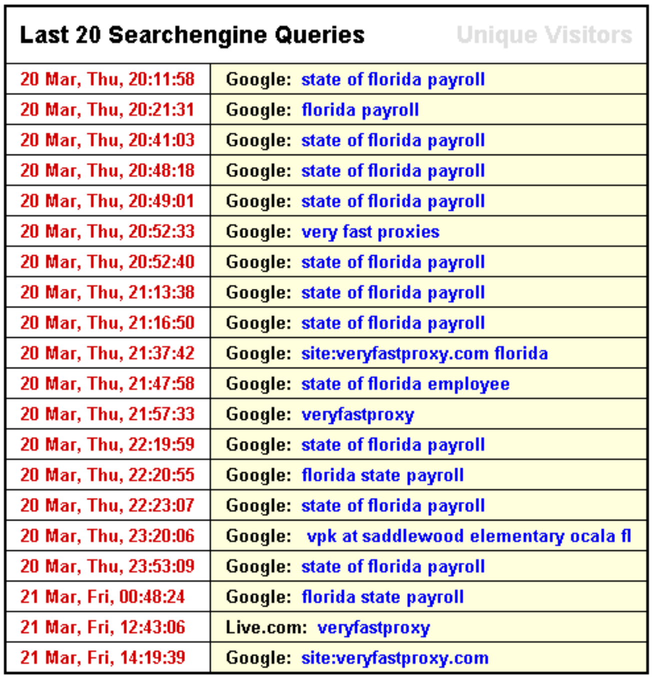
In March 2008, Florida state employees used a German Web proxy to bypass the state's AUP, prompting fears that some may have used the server to access the Florida Accounting Information Resource (FLAIR) system. This prompted a precautionary statewide password reset (Tallahassee Democrat, 2008).

After claiming they had broken "all links with known proxy servers" and declaring that "no one could hack" the FLAIR system, Florida state employees were still using the German proxy, as shown below from the site's FastTracker Web stats.



Last 20 Visitors	Unique Visitors
25 Mar, Tue, 00:50:45	adv173.neoplus.adsl.tpnet.pl
25 Mar, Tue, 01:46:53	c-76-109-249-162.hsd1.fl.comcast.net
25 Mar, Tue, 03:09:29	adsl-66-142-213-247.dsl.tpkaks.swbell.net
25 Mar, Tue, 04:50:26	c-68-60-107-224.hsd1.mi.comcast.net
25 Mar, Tue, 07:32:50	cache-mex-vallejo-1.prodigy.net.mx
25 Mar, Tue, 07:49:25	host-89-228-28-80.zamosc.mm.pl
25 Mar, Tue, 08:49:10	host-static-89-41-70-41.moldtelecom.md
25 Mar, Tue, 09:36:08	6.ver.airbites.pl
25 Mar, Tue, 09:56:30	net-136-105.tarman.pl
25 Mar, Tue, 10:23:42	tp98.internetdsl.tpnet.pl
25 Mar, Tue, 11:15:15	audsem01.aud.state.fl.us
25 Mar, Tue, 11:20:28	205-186.tepsalan.net.pl
25 Mar, Tue, 11:25:23	77-45-42-186.sta.asta-net.com.pl
25 Mar, Tue, 11:46:26	199.73.152.100
25 Mar, Tue, 12:38:51	dohspxy1.doh.state.fl.us
25 Mar, Tue, 12:42:37	190.129.204.68.cfl.res.rr.com
25 Mar, Tue, 12:45:02	mobile-166-214-184-087.mycingular.net
25 Mar, Tue, 12:48:00	199.73.152.100
25 Mar, Tue, 13:05:35	sdrblue1.dor.state.fl.us
25 Mar, Tue, 13:05:52	host1.esw.pl

Also from the FastTracker site statistics, it was notable that Internet searches for "Florida Payroll" were landing at the German proxy site itself!



Last 20 Searchengine Queries	Unique Visitors
20 Mar, Thu, 20:11:58	Google: state of florida payroll
20 Mar, Thu, 20:21:31	Google: florida payroll
20 Mar, Thu, 20:41:03	Google: state of florida payroll
20 Mar, Thu, 20:48:18	Google: state of florida payroll
20 Mar, Thu, 20:49:01	Google: state of florida payroll
20 Mar, Thu, 20:52:33	Google: very fast proxies
20 Mar, Thu, 20:52:40	Google: state of florida payroll
20 Mar, Thu, 21:13:38	Google: state of florida payroll
20 Mar, Thu, 21:16:50	Google: state of florida payroll
20 Mar, Thu, 21:37:42	Google: site:veryfastproxy.com florida
20 Mar, Thu, 21:47:58	Google: state of florida employee
20 Mar, Thu, 21:57:33	Google: veryfastproxy
20 Mar, Thu, 22:19:59	Google: state of florida payroll
20 Mar, Thu, 22:20:55	Google: florida state payroll
20 Mar, Thu, 22:23:07	Google: state of florida payroll
20 Mar, Thu, 23:20:06	Google: vpk at saddlewood elementary ocala fl
20 Mar, Thu, 23:53:09	Google: state of florida payroll
21 Mar, Fri, 00:48:24	Google: florida state payroll
21 Mar, Fri, 12:43:06	Live.com: veryfastproxy
21 Mar, Fri, 14:19:39	Google: site:veryfastproxy.com

Although “unhackable”, the state’s payroll system drew a great deal of unwanted attention from someone. This incident demonstrates how employee usage of anonymous proxies can waste time and money across an entire organization.

1.2.3. “I got pwned and blacklisted, now what?!”

Also in 2008, an Illinois school technology coordinator sought to “help out” his students by removing the username and password requirement for the school’s proxy server, which had a public IP address and an Internet content filter (O'Hagan, 2008).

In his own words,

“My thought was why would someone use a proxy that has a filter on it?”

The school’s network was soon overwhelmed with traffic from around the world. Much of it was SPAM and resulted in the school’s public IP address being banned as a known SPAM site.

This incident demonstrates that once an open proxy is listed online, it is rapidly overwhelmed by proxy-hungry flashmobs. This is so common with open proxies that a proxy that stays up for more than a few days is a rare find.

Misconfiguring a proxy server in this way is also a serious issue for the network on which the proxy server resides, since the proxy will also satisfy requests directed to the inside as well as the outside. Skilled hackers have long been known to leverage these common proxy configuration mistakes for intrusion activities (Mitnick & Simon, 2005).

2. Proxy List Research

Research began in March 2008 and continued for over thirty months. Over 350 public proxy lists were scanned for new entries daily, with the most active sites on a bihourly schedule.

2.1. Data Collection Tools

The proxy list research project was developed and run on a dedicated Xubuntu 8.04 (and later, 9.04) virtual machine (VM) on VMWare's GSX Server v1.0.10 running on top of Microsoft Windows XP. Internet connectivity was supplied by Insight RoadRunner of Columbus, Ohio and the project was designed to run within the constraints of Insight's Terms of Service.

2.1.1. Data Retrieval

All data retrieval of proxy lists was performed with bash scripts and Open Source tools, including:

wget – the standard tool for retrieving simple Web pages, wget allows the user to specify a proxy on the command line, provide a false User-Agent (some proxy lists block the default wget User-Agent) and other important headers.

curl – specifically desirable when a Web page requires a POST rather than a GET to download the page. Proxy can be specified on the command line.

netcat – sometimes required when cookies need to be retrieved and sent back to the page to get a result. Not designed to be proxy-friendly, netcat is versatile enough for it not to matter.

links2 – an HTML to text utility, used for difficult pages that don't script well.

html2txt – another Web-to-text utility. Although useful for translating Unicode, this tool has the unfortunate habit of outputting extremely strange "text" in the format of *[char1][backspace][char1]*, but this can usually be eliminated by post processing the retrieved page.

2.1.2. De-Obfuscation

The proxy lister is often jealous of his data and often deploys countermeasures to prevent page-scraping, which is widespread. A common method is to deny access based on the default User-Agent values of the Web scripting utilities wget and curl. One memorable site automatically returned "I'm going to report you to your ISP you little jerk!" whenever accessed via wget. However since both curl and wget allow the user to specify a custom User-Agent this method is ineffective against experienced list scrapers.

With the exception of AJAX-based pages (to date, very rare), none of these methods are very effective and the following Open Source tools have been found to be useful in bypassing them.

rhino – provided by the Mozilla project, rhino is a Javascript interpreter written in Java, allowing the user to execute Javascript on the command line, and hence in scripts. Javascript obfuscation is very common in proxy lists, but it is often simpler to de-obfuscate than HTML itself.

gocr – GNU Optical Character Recognition. A few of the more active proxy list sites attempt to obfuscate their pages by displaying *images* (typically in GIF format) of text instead of the text itself. A little rough around the edges, gocr will often confuse a number with a letter but standard post-processing of the output (searching/replacing "B" for "8", "S" for "5", and changing all instances of the letter "O" to zero, for instance) provides excellent, reliable results.

2.1.3. Miscellaneous Tools

In addition to the above, the following tools were required.

mysql – a database was required to store and track the distribution of proxies and the date of their appearance. MySQL includes commands to convert IP addresses between their dotted-decimal format and their 32-bit integer values, a required feature not always available in other databases.

GeoLite City – the free version of MaxMind's's excellent geoIP database was the only non-Open Source tool used during the project.

Google – indispensable for locating proxy lists, search results were often surprising as well as informative.

nmap – used for initial testing of reported proxy servers.

grep – a workhorse utility leveraged for HTML-scraping.

sed – required for slicing, dicing, and general formatting of harvested Web page data in preparation for insertion into the MySQL database.

2.1.4. The ASP Proxy Judge

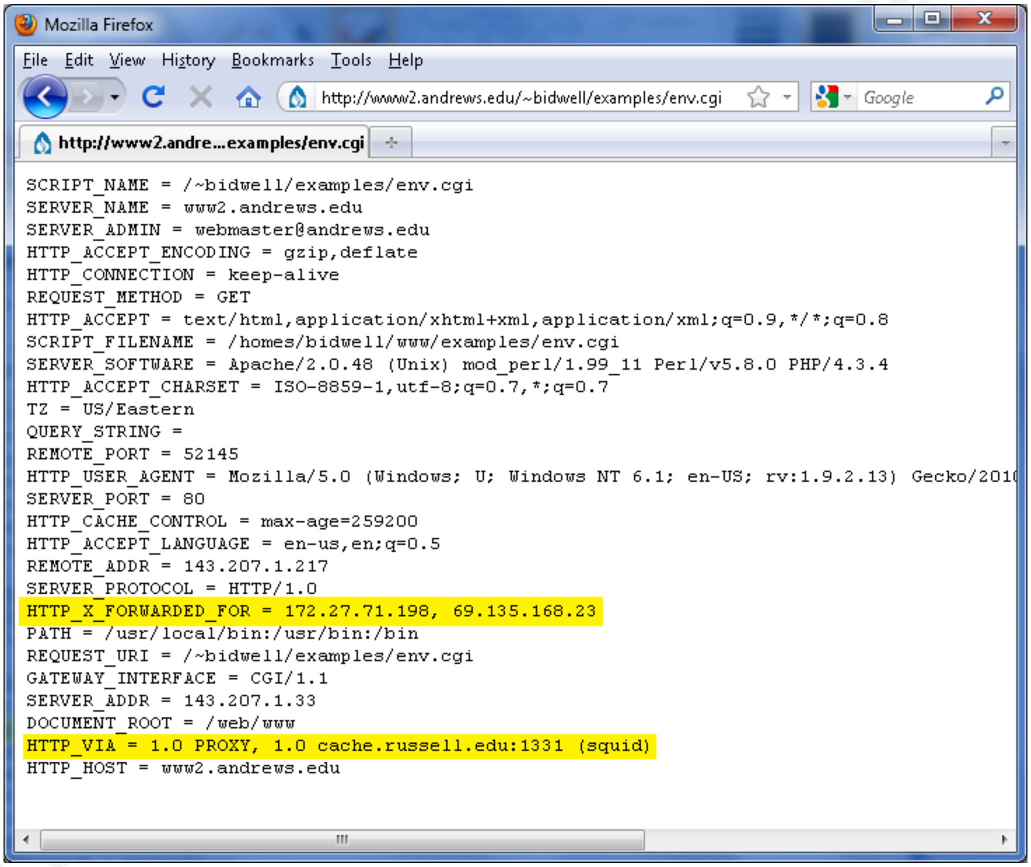
The single most important tool for classifying proxies is the proxy judge. This is a Web page that will return the HTTP headers set by the proxy when connecting to a remote site. By analyzing the returned headers the proxy is rated for the degree of anonymity offered. The basic decision matrix is shown below.

	Transparent	Anonymous	High Anon
X-Forwarded-For	✓		
Via		✓	
None			✓

When accessing a proxy judge through a "Transparent" proxy the HTTP request always include the X-Forwarded-For header, which is the IP address of the user. Transparent proxies do not allow privacy or anonymity since the user's address is always revealed to the remote site. In practice, this value is seldom logged at the remote site, so it is up to the proxy user to decide whether a transparent proxy is safe enough for his activities.

An “Anonymous” proxy does not include the X-Forwarded-For header but will return a Via header, which identifies the *proxy* rather than the user. All HTTP 1.1 complaint proxy servers are required to include this header (Fielding, et al., 1999).

A high anonymous (High Anon) proxy server has neither header, making it indistinguishable from an HTTP request made by a normal user over a direct route. Activity cannot be traced back to the user (transparent) or the intermediary proxy (anonymous).


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www2.andrews.edu/~bidwell/examples/env.cgi
Google
http://www2.andre...examples/env.cgi

```
SCRIPT_NAME = /~bidwell/examples/env.cgi
SERVER_NAME = www2.andrews.edu
SERVER_ADMIN = webmaster@andrews.edu
HTTP_ACCEPT_ENCODING = gzip,deflate
HTTP_CONNECTION = keep-alive
REQUEST_METHOD = GET
HTTP_ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
SCRIPT_FILENAME = /homes/bidwell/www/examples/env.cgi
SERVER_SOFTWARE = Apache/2.0.48 (Unix) mod_perl/1.99_11 Perl/v5.8.0 PHP/4.3.4
HTTP_ACCEPT_CHARSET = ISO-8859-1,utf-8;q=0.7,*;q=0.7
TZ = US/Eastern
QUERY_STRING =
REMOTE_PORT = 52145
HTTP_USER_AGENT = Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.13) Gecko/201
SERVER_PORT = 80
HTTP_CACHE_CONTROL = max-age=259200
HTTP_ACCEPT_LANGUAGE = en-us,en;q=0.5
REMOTE_ADDR = 143.207.1.217
SERVER_PROTOCOL = HTTP/1.0
HTTP_X_FORWARDED_FOR = 172.27.71.198, 69.135.168.23
PATH = /usr/local/bin:/usr/bin:/bin
REQUEST_URI = /~bidwell/examples/env.cgi
GATEWAY_INTERFACE = CGI/1.1
SERVER_ADDR = 143.207.1.33
DOCUMENT_ROOT = /web/www
HTTP_VIA = 1.0 PROXY, 1.0 cache.russell.edu:1331 (squid)
HTTP_HOST = www2.andrews.edu
```

Typical proxy judge page showing Via and X-Forwarded-For headers

Users who require strict anonymity typically prize High Anon proxy servers over the Anonymous and Transparent varieties.

A Google search for “proxyjudge” will return over 125,000 results, including a wide variety of free and for sale software, lists of online proxy judge Web pages, proxy forum chatter on the subject, various HOWTOs, and the usual smattering of SEO optimized link farms looking for ad revenue and click-throughs.

Examining these search results, it is evident that proxy judging is an industry-within-an-industry. Running a proxy judge Web page also offers the proxy list compiler a passive tool for the collection of proxy data. Many proxy listers will run just such a page for this purpose, mining the data in their Web site logs for connections from open proxies. However, since these pages are heavily used, their availability is spotty and most sites limit connections per day to prevent abuse. To host such a Web page and advertise it aggressively is to invite a good deal of automated proxy test traffic and unintended Denial of Service.

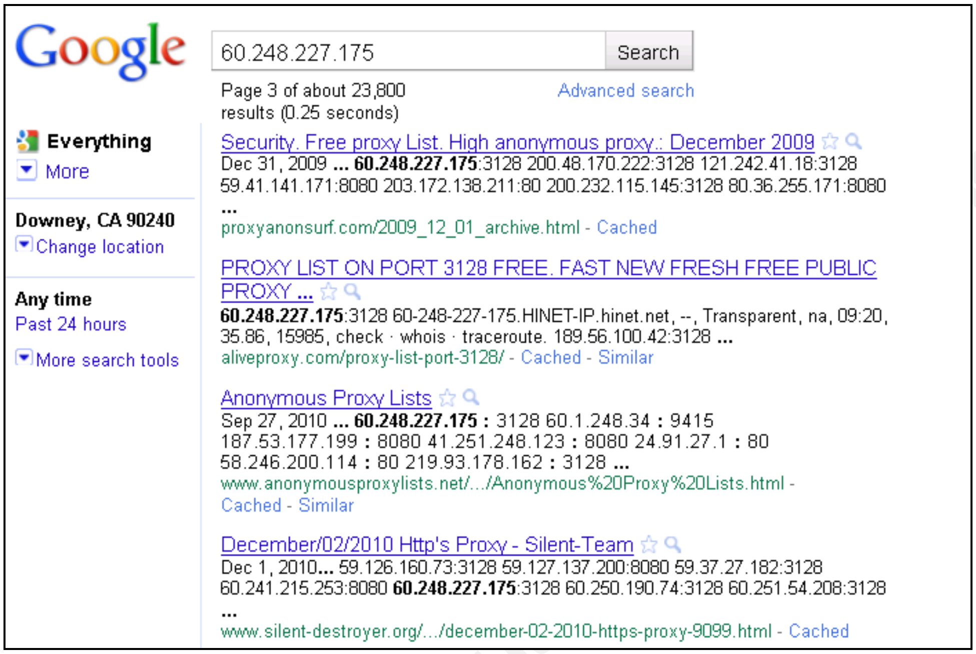
Fortunately, this service is also offered by a number of legitimate sites that are entirely unaware they are promoting proxy testing. In the early days of Active Server Pages (ASP) 1.0, Microsoft published a sample ASP script, *servervariables.asp*, that performs the exact function of a proxy judge. A simple Web search will identify hundreds of these pages. Over the years, this script was often cut and pasted onto Web servers during testing or deployment and forgotten. It has also found its way into dozens of ASP programming sites, customized with colors and neatly formatted tables. It is never advertised as a proxy judge and since it resides on numerous established but lonely and neglected Web servers the traffic is low and the availability is high. Approximately fifty of these URLs were collected and used for the purpose of this research.

Hosting such a page is probably ill-advised, but fortunately the majority of proxy testers hammer the most popular sites, such as proxydetect.com, a passive detector site based in Moscow, Russia and related to longtime proxy lister and “privacy advocate” AtomIntersoft (http://atomintersoft.com/).

2.2. Data Gathering Methodology

2.2.1. Establishing a Baseline

In order to track the growth and distribution of proxies, it is necessary to establish a historical baseline to compare new data to as it is published to proxy lists and subsequently acquired for research. One of the first lessons learned during the early months of the project in 2008 was that *all public proxy lists contain nearly identical data*. A single published proxy IP address can be found on literally thousands of proxy list sites. Take, for example, the IP address 60.248.227.175, which first appeared on proxy lists in May 2009.



60.248.227.175

Page 3 of about 23,800 results (0.25 seconds)

Security. Free proxy List. High anonymous proxy.: December 2009
Dec 31, 2009 ... **60.248.227.175**:3128 200.48.170.222:3128 121.242.41.18:3128 59.41.141.171:8080 203.172.138.211:80 200.232.115.145:3128 80.36.255.171:8080 ...
proxyanonsurf.com/2009_12_01_archive.html - Cached

PROXY LIST ON PORT 3128 FREE. FAST NEW FRESH FREE PUBLIC PROXY ...
60.248.227.175:3128 60-248-227-175.HINET-IP.hinet.net, --, Transparent, na, 09:20, 35.86, 15985, check · whois · traceroute. 189.56.100.42:3128 ...
aliveproxy.com/proxy-list-port-3128/ - Cached - Similar

Anonymous Proxy Lists
Sep 27, 2010 ... **60.248.227.175 :** 3128 60.1.248.34 **:** 9415 187.53.177.199 **:** 8080 41.251.248.123 **:** 8080 24.91.27.1 **:** 80 58.246.200.114 **:** 80 219.93.178.162 **:** 3128 ...
www.anonymousproxylists.net/.../Anonymous%20Proxy%20Lists.html - Cached - Similar

December/02/2010 Http's Proxy - Silent-Team
Dec 1, 2010... 59.126.160.73:3128 59.127.137.200:8080 59.37.27.182:3128 60.241.215.253:8080 **60.248.227.175**:3128 60.250.190.74:3128 60.251.54.208:3128 ...
www.silent-destroyer.org/.../december-02-2010-https-proxy-9099.html - Cached

Typical Google search for a well-known proxy IP address

A Google search of this (Chinese) IP address reveals it shows up on no less than 23,000 pages. Not all of these results are proxy related sites; some are abuse reports and some are network information sites.

Obviously, a solid database of known proxies would be required to track growth. This took about four months of collecting data from the most active proxy lists and forums and one of several serendipitous Google findings: a spammer's proxy collection. The site *newcenturyweightloss.net* (now defunct, but cached at http://web.archive.org/web/20080703022756/http:/newcenturyweightloss.net/), discovered in 2008, contained data on over 800,000 proxies tucked away in .txt and .rar files.

Another lucky find in 2008 was *drochim.com*, which is a domain name once registered to notorious fraudster Nikolai Lidiaev (McQuaid, 2010) and hosted in a SoftLayer Technologies (at one time a Russian Business Network "affiliate", but now owned by GoDaddy) netblock. At the time, *drochim.com* was an active, but largely unadvertised, proxy site (it has since changed hands and host providers and now specializes in online pornography). Several obfuscated links at the site delivered proxy address/port combinations that updated in real time, perhaps indicating botnet activity. This site contributed tens of thousands of proxy IP addresses until it disappeared in the Fall of 2008, a likely casualty in the EST Domains/Intercage ISP takedowns during that time period (Danchev, 2008).

Another, more public, site with Russian ties, awmproxy.com (a Russian language site registered in the Netherlands and hosted in Germany), which is still in existence, distributed proxy addresses to paying customers through an insecure system of obfuscated URLs. Although obfuscated, they were not immune to GoogleBot indexing and ten of thousands of proxy addresses reserved for paying customers were leaked out to the Web. Once these URLs were discovered and widely distributed to proxy forums, the administrators of *awmproxy.com* found a more secure method to distribute their wares and the source dried up.

These three sites alone contributed at least a third of the three million proxies that were collected during the thirty months of the project. The majority of the data came from day-to-day harvesting of data from the most active proxy sites, as well as occasional Google deep-dives for additional data. In fact, *newcenturyweightloss.net*, *drochim.com*, and *awmproxy.com* were accidental discoveries made during these same deep-diving missions. At first, custom scripts were designed for the popular sites, but as these diving exercises progressed the approach and coding style became more generalized to the extent that most sites that use an unobfuscated, dotted-decimal, tabular proxy/port format could be assimilated with the same code. Site URLs harvested from Google were placed in their own database and re-scraped once a day. The popular, active sites were revisited every two hours. Results were published to the Web between these runs.

2.2.2. Collecting & Compiling Results

Collection of proxy server data was performed in scheduled batch runs throughout the day, hitting the most active sites bihourly and a database of less active sites daily. Proxy address and port data was consolidated in text files, sorted, and duplicate entries removed.

Before being stored in the database, each address and port combination was checked against it to ensure it wasn't a duplicate. If it was, it was immediately discarded. On discovery of a newly published proxy that did not already exist in the database, the listed port was tested with nmap, with three possible outcomes:

- The port was open.
- The port was closed.
- The port test timed out (45 second maximum).

After the port test, reverse DNS and geoIP lookups were performed, the time and date collected, an SQL INSERT statement created, and the master proxy database updated.

After the initial entry into the database, a second bihourly run was scheduled to retrieve a random ASP proxy judge page from proxies found to be listening (listed port open). On successful retrieval of the page, the proxy was classified into one of CoDeeN, Transparent, Anonymous, or High Anonymous, as determined by the HTTP headers revealed by the proxy judge.

CoDeeN servers represent a special case. To detect CoDeeN servers, it is necessary to create a unique HTTP User-Agent for each request. CoDeeN tracks users by logging a unique identifier based on the remote user's IP address and User-Agent (Web browser version). An IP address and User-Agent combination not previously seen by the network in a given time period will receive the standard CoDeeN greeting—the desired result—whereas established users—those whose IP and User-Agent *have* been previously collected—pass through to the destination site without the greeting. This essentially renders the proxy judge results useless.

For example, a typical User-Agent such as the following standard Internet Explorer 7.0 agent…

```
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
```

…can be sufficiently randomized for CoDeeN "evasion" (a misnomer in the sense we *want* to be detected as a new user) by inserting random digits into the NET CLR (.Net Framework Common Language Runtime) or other version numbers.

Once these data have been collected, the results are inserted into a separate table of live proxies in the database. After a successful live run, both the live proxy table and the master proxy table are purged of any duplicates, maintaining the earliest record of discovery.

3. Results

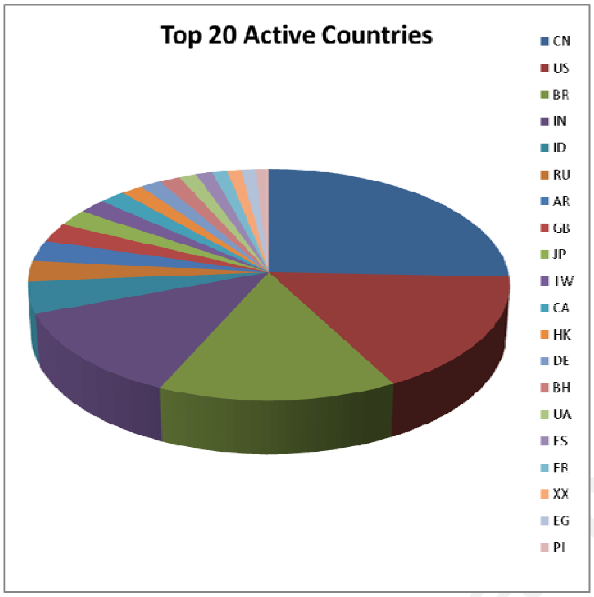
The method described above harvested an average of 100,000 proxies per month over the course of thirty-two months, for a total of over three million unique entries.

3.1. General

Today's casual proxy hunter has a difficult, frustrating job ahead of him. He has to navigate through an incredible maze of search results to find an active proxy list. Once found, his chances of finding a live proxy are close to zero. On any given public proxy list, a 1% success rate is common. 4% is a veritable gold mine. The data agree. Of the three million proxies tested, a mere 5% (~170,000) presented open ports at the time of their discovery.

With this kind of luck, he will likely take his quest to the many proxy forums where users share their proxy hunting trophies, only to find they are scraping the same proxy lists he has just abandoned.

If he's having a good day he may find a handful of live addresses. Chances are, if he bothers to look, today that proxy would be located in Brazil, Indonesia, the US, India, China, or Russia. If he bothers to check by reverse DNS lookup, the IP address will be most likely belong to a residential broadband customer (for example, a DNS name ending in *.socal.res.rr.com* – over 600,000 *.rr.com* addresses can be found in the master proxy database).



In fact he would be hard-pressed *not* to find a proxy on a residential broadband account. That is where all the action is. True, the occasional advanced home user may enable a proxy on her account for personal use, but the majority of these proxies are the result of system compromise. As we will see, residential broadband proxies can be used as a reliable barometer and predictor of malware spread.

If the proxy hunter decides not to use a broadband proxy, his choices are severely limited. Relatively very few (~12,000) proxies disclose the fact they *are* proxies in their DNS names, for example:

proxy.genevaonline.com
proxy.bezeqint.net
proxy1.springer.de
proxy1.iunet.it
proxy.ledger-bennett.co.uk
proxy.kds.de
proxyh.emirates.net.ae
proxy.logos.cy.net
proxy2.pironet-ndh.com
proxy.ms.tlk.com
proxy.apf.it

Most of these legitimate proxy servers eventually filter access to their own customer base to prevent abuse.

A similar number (~10,000) of proxies have been found on mail servers, *i.e.*, servers with DNS names that start with "mail". These also tend to be filtered against abuse.

3.2. Proxy Consumers

In order to determine *who* the primary consumers of proxy lists were, the research list was published to the Web bihourly, between batches of proxy list harvests. Care was taken to inform the casual user that the list was in fact a security research project, proxy use was dangerous and not recommended, and in certain countries users could face legal penalties for unauthorized use of open proxies. Beginning in June 2008, usage was tracked by Extreme Tracking's (http://extremetracking.com/) free utility, aptly named FreeTracker, a small snippet of JavaScript combined with a webbug graphic that records the amount and source of traffic to the page it is embedded in.

Although FreeTracker has issues with information disclosure—anyone clicking on the webbug can view the IP address and other details of the most recent visitors, as we learned from the State of Florida—it was chosen over the host provider's services simply for the amount of detail it provides.

The page experienced a very modest hit rate of 350-500 unique users per day for a lifetime total of over 440,000. The average list user (98%) ran some form of Windows OS—mostly (86%) XP—using Firefox 3.*x* (70%) with JavaScript enabled (99%). Oftentimes in the beginning he landed on the page when searching for a specific IP address or a search expression like "proxy server in Brussels" but as time went by most users simply searched for the list by name.

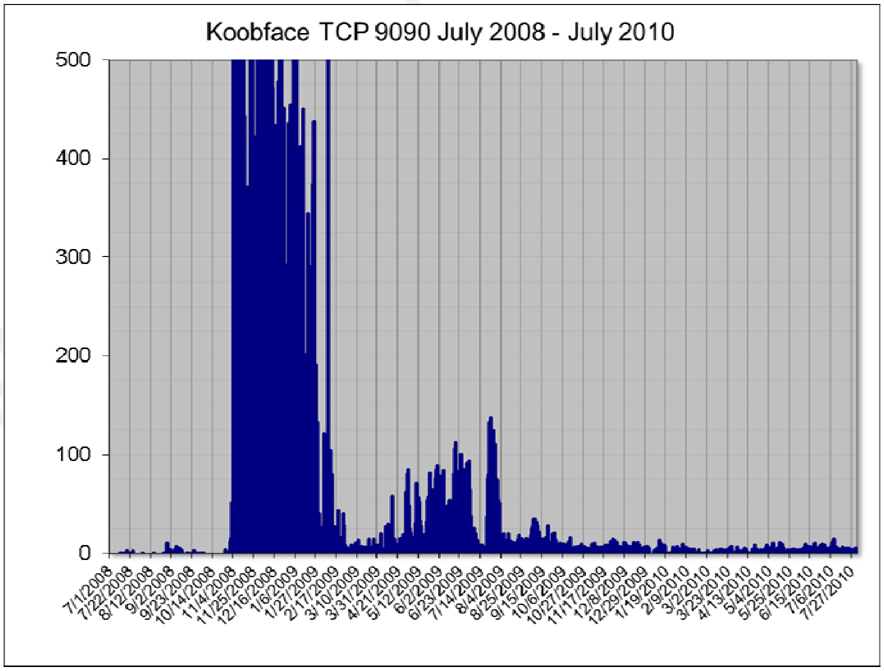
But the most telling of these otherwise bland statistics was the average user's country of origin. Nearly two-thirds of the list's users (62%) were located in Cameroon, an African nation due south of the infamous home of 419 scams, Nigeria, which itself ranked a distant 16th place.

After mentioning this in a blog post, the author was contacted, in blog comments, by a small number of Cameroonian scam artists, going by monikers such as Pappa Dollars, STARVO, and Dableed, pleading for proxies located in the United Kingdom, a country that treated Cameroon poorly after acquiring the former Germany colony in 1919. Although the country was divided between Britain and France in 1919 (US State Dept., 2010), many Cameroonians despise the UK intensely and have singled them out in countless online “Puppy Scams” over the years (Kahn, 2010). UK based proxies allow them to mask their location to effectively pull off these scams. Fortunately for the UK, their security awareness has improved and, until the Koobface Trojan spread of 2009, English proxies were relatively rare.

Oddly, for as much as proxies are touted on the Web as tools for promoting freedom and evading censorship (Wikipedia, 2002), countries where these benefits would be appreciated most (China, Iran, the UAE, *etc.*) were conspicuously underrepresented (CPJ, 2006), although this too could be an indirect indicator of censorship.

3.3. Koobface(TinyProxy) TCP Port 9090

First detected in August 2008, W32.Koobface was considered a low risk Trojan by security vendor Symantec (Chien & Shearer, 2008).



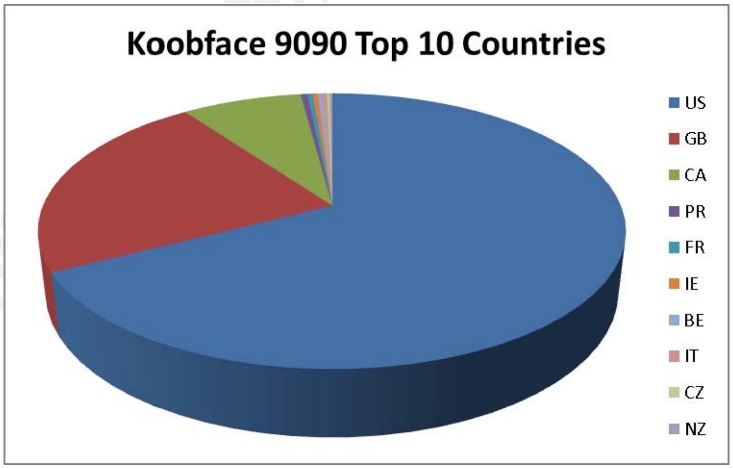
Prior to that announcement, there were barely 20 listed proxies using TCP port 9090, but by the end of the month the total had more than tripled.

By September 4th, it was a daily occurrence. In October, it was conspicuously absent. On November 1st, as if on schedule, the floodgates opened, and daily counts spiked to over 1300 in less than two weeks. Concurrently, users were starting to report infections of Backdoor.TinyProxy. By the 18th, removal instructions were published (SpywareRemove, 2008).

Although a Trojan in its own right, TinyProxy turned out to be a component, one of many, of Koobface.

On December 4th, the day *after* reports of a "new Koobface infection" hit the press (Vamosi, 2008) daily port 9090 proxies peaked at over 1800. This continued through January of 2009, dropping off in February and March, thereafter enjoying a "dead cat bounce" in the Spring and Summer of that year (as seen in the chart above), after which it dissolved into background noise, but not before being declared the Web's largest botnet (Baltazar, Costoya, & Flores, 2009).

This Koobface variant was a plague on the English-speaking world. 97% of all victims were located in the US (66.6%), Great Britain (22.3%), and Canada (7.8%).

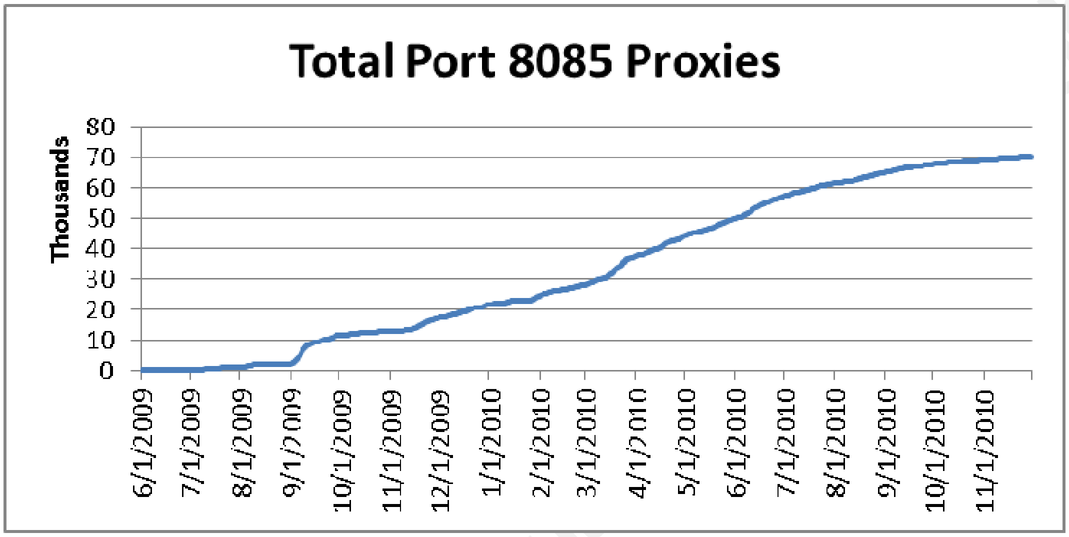


Then next Koobface variant would narrow its focus.

3.4. Koobface (SillyProxy) TCP Port 8085

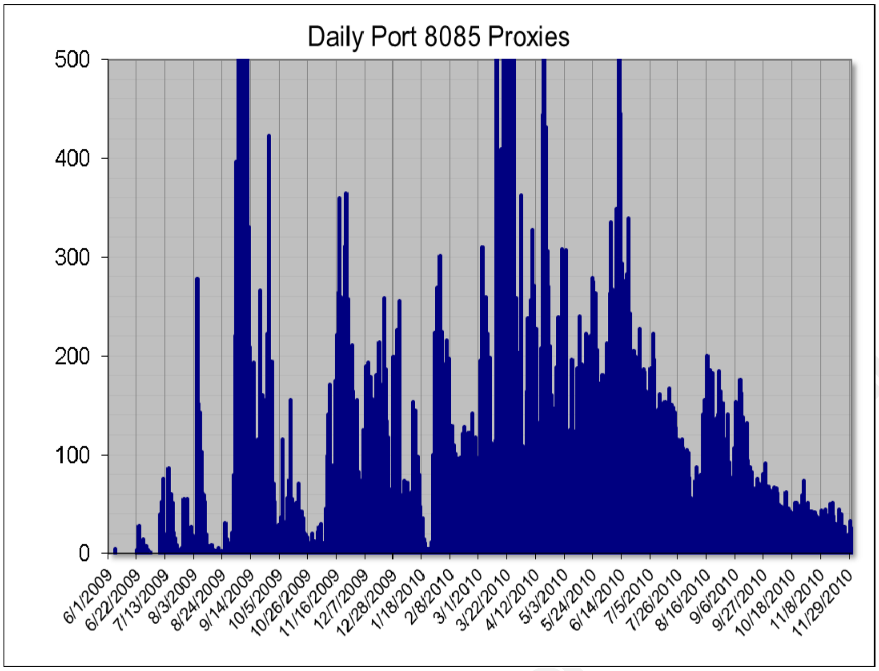
TCP port 8085 was another seldom seen port in proxy lists. It made its debut in November 2008, but fell off the lists until June of 2009, when it first came to the attention of an independent security researcher (Unknown, 2009). By July 2nd, Computer

Associates had labeled it Win32/SillyProxy.DG (Robielos, 2009), documenting its use as a proxy server for "Koobface variants". SillyProxy was seen as far back as 2006 (Computer Associates, 2006), but whether it was using TCP port 8085 at that time was not documented.



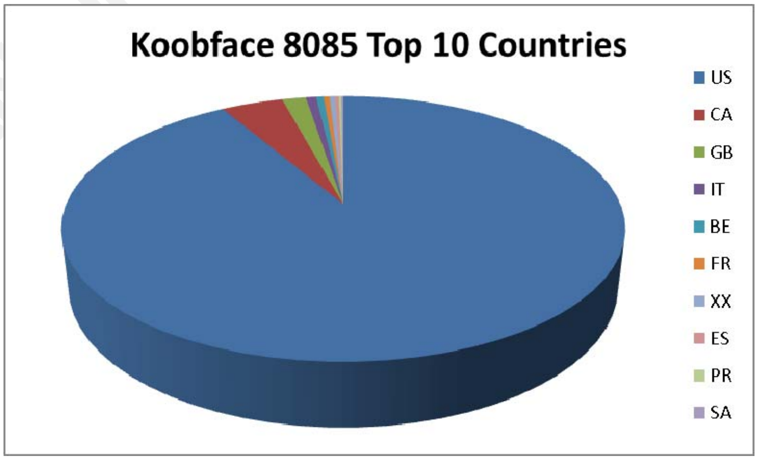
As the chart above demonstrates, port 8085 has enjoyed steady growth ever since and has only recently begun to level off, mirroring the growth of Facebook (White, 2010).

By September 2009, Microsoft had named two variants of the port 8085 version of Koobface, Win32/Koobface.gen!F (Microsoft, 2009) and Win32/Koobface.gen!Q. (Microsoft, 2009) although other researchers had claimed more (itwire, 2009). The chart below graphs the number of port 8085 proxies appearing daily in proxy lists after these discoveries and roughly identifies three different surges: June-October 2009, November 2009-January 2010, and February-August 2010.



In all, this variant was quite successful for well over a year and reportedly earned its creators over $2M US during its lifetime, primarily from the sale of fake anti-virus products commonly known as “scareware”. The command and control centers were finally taken down in November 2010 (Deibert & Rohozinski, 2010).

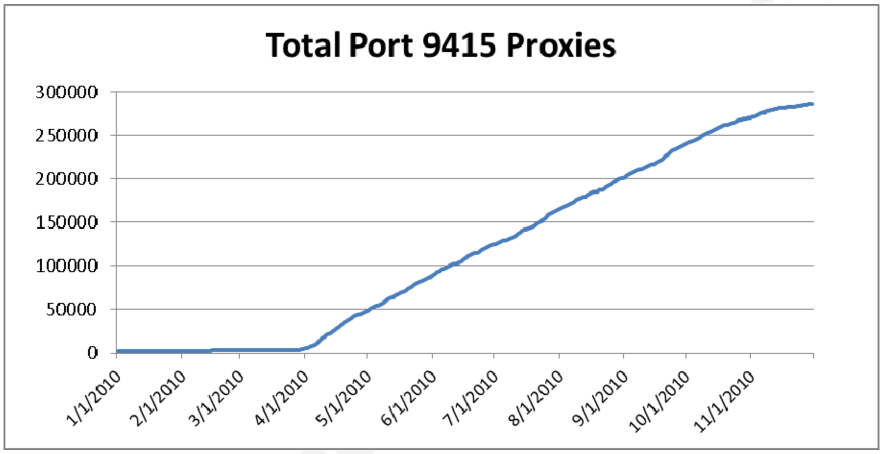
Unlike its predecessor, this version flourished in the U.S.A., with 90% of all port 8085 proxies.



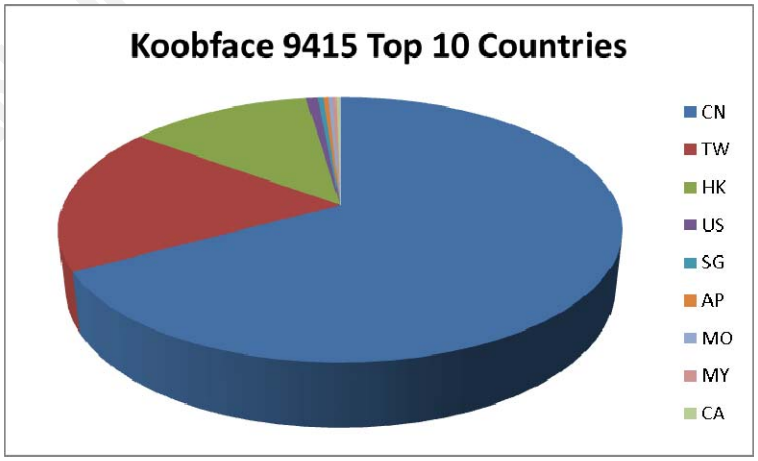
3.5. Koobface China – TCP Port 9415

While Koobface 8085 was making a mess in the Fall & Winter of 2009, another new port, TCP 9415, was starting to show up on proxy lists across the Web. Off to a slow start in 2008, total numbers had finally reached triple-digit status in mid-September 2009. By January 1st of 2010 there were well over 2000.

By the end of the year there would be over a quarter of a million, as shown in the chart below.

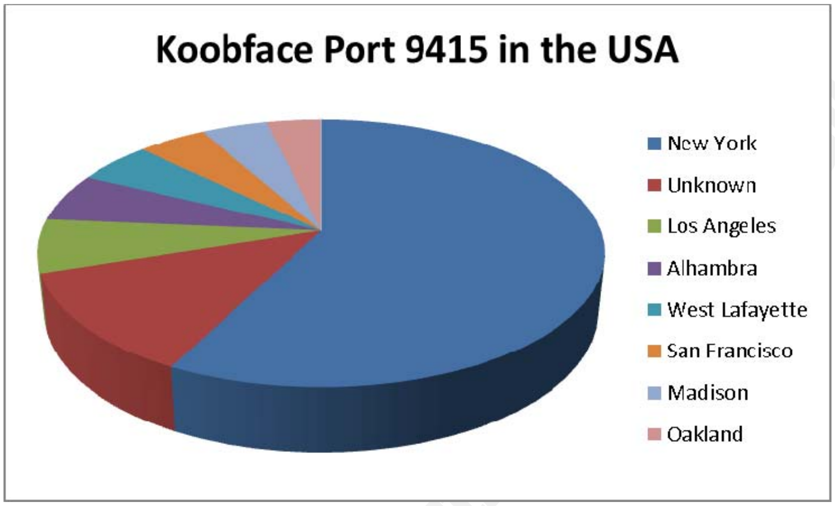


Before it was confirmed as a Koobface variant, it was evident by sheer numbers that Chinese-speaking countries had a virtual monopoly on port 9415 proxies.

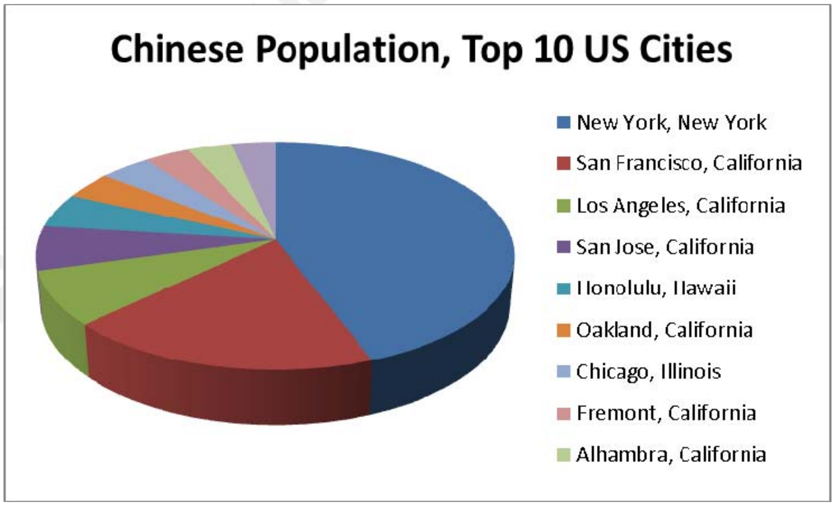


In August of 2010, well into the spread of the systems, the true story was finally revealed. A Chinese language instant messaging (IM) client, Tencent QQ, had been targeted by malware served by Network Solutions Web sites (Naraine, 2010) and parked pages (default pages for domains that have no active content) (NetSol, 2010). Its success was further guaranteed by its near immunity to anti-virus software (Krebs, 2010).

The success of the 9415 bug in the United States is revealed by a GeoCity Lite analysis of the infection.



The distribution corresponds closely to the distribution of Chinese people in the United States (Wikipedia, 2004).



. GeoCity Lite was unable to place IP addresses in the "Unknown" slice, but an analysis of the resolved host names, most of which are once again residential broadband customers, reveals hosts from Hawaii, New York City, and California.

This host name check also identifies why West Lafayette, Indiana is *not* in the list of the top ten cities with significant Chinese population: the addresses were overwhelmingly registered to Purdue University, which has enjoyed increased Chinese student enrollment in the past few years (AP, 2008).

Since Chinese speaking users are likely to favor Chinese language IM clients, it is clear these U.S. infections shared the same infection vector, Tencent QQ.

Although the spread of this bug was larger than the combined spread of its predecessors, it garnered little press attention, aside from Network Solutions' security problems.

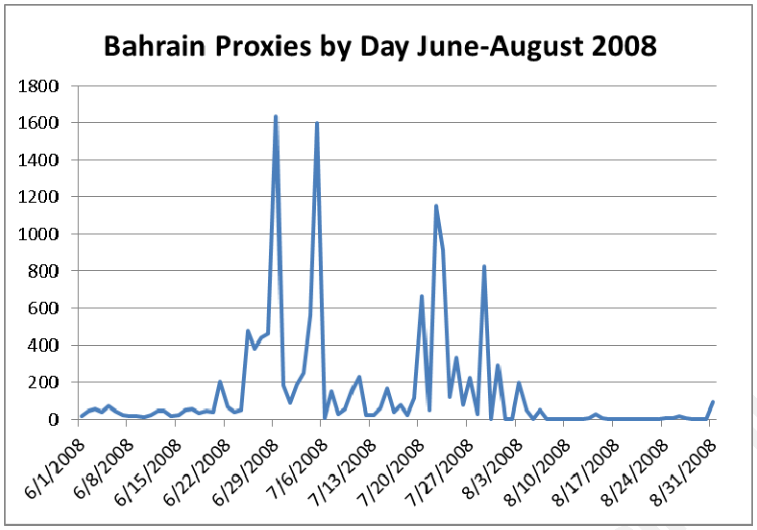
3.6. Proxy Hot Spots and Microbursts, 2008-2010

Although as a general rule proxies tend to follow the population, there have been notable incidents of growth identified in cities and institutions across the globe that can't be directly attributed to the spread of malware.

3.6.1. Bahrain 2008

Tiny Bahrain, a wealthy Arab island nation located in the Persian Gulf, has suffered from Internet censorship since 2005, when all Web traffic was routed through a government-controlled proxy in an effort to block sites deemed to be anti-Islamic or anti-government (Hanford III, 2006) and administered by the country's telecommunications monopoly, Bahrain Telecommunications Company (*a.k.a.* Batelco). In 2007, Batelco changed their censorship infrastructure: instead of centralized proxies, censorship was moved closer to the user, into consumer routers, briefly unblocking previously censored sites during the changeover (Al Yousif, 2007).

This meant Batelco now had to maintain the block lists on thousands of routers remotely. In 2008, an improper mass reconfiguration opened up the proxy capabilities of these devices to the world and the daily count of open proxies in Bahrain skyrocketed, spiking to over 1600 per day at one point.

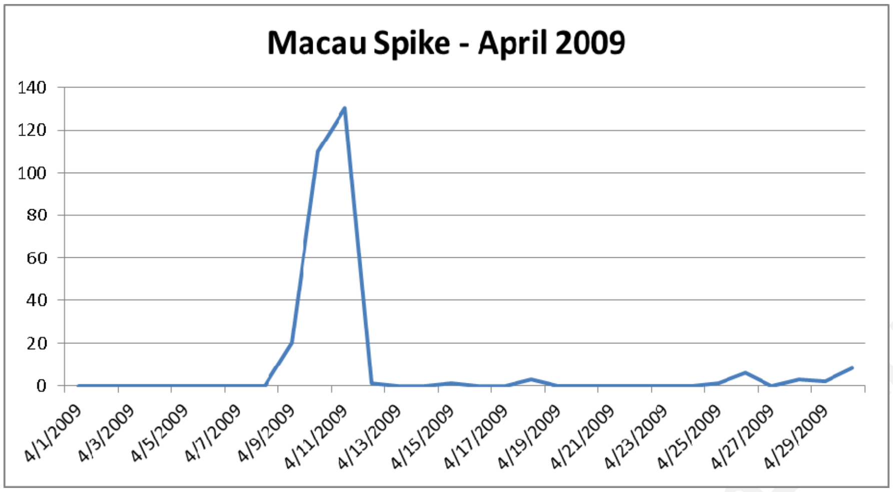


In all, over 17,000 proxies were seen, but by September the flood was over and they were never seen again. In fact Bahrain escaped the subsequent Koobface epidemics, with only a single address found to be listening on a known Koobface port in January 2009, even though Trend Micro had claimed it had cleaned 159,228 infected systems in Bahrain in the first three months of the year (TradeArabia, 2009).

3.6.2. Macau 2009

Situated across the Pearl River Delta from Hong Kong, Macau shares a similar history with Hong Kong. Once a Portuguese colony, it was handed over to Chinese control in 1999 along with Hong Kong (Wikipedia, 2003). Internet broadband services are provided in Macau by Companhia de Telecomunicacoes de Macau (CTM).

In April 2009, CTM made an administrative mistake similar to Batelco's 2008 router misconfiguration. Although not as dramatic as Batelco's move, beginning in early April 2009, CTM exposed the proxy capabilities of its customers' routers to the Internet at large. This did not go unnoticed by the proxy scanners and listers of the world and by the 11th over 300 proxies were available.

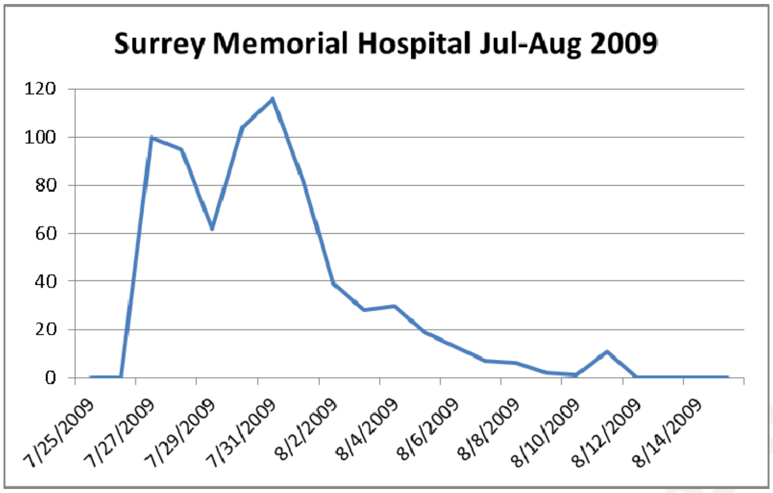


But no sooner had they appeared the mistake was repaired and the proxies went offline. Relatively few Macau proxies were ever seen again, until the Koobface 9415 variant hit China at large and nearly 900 CTM customers were infected.

3.6.3. Surrey Memorial Hospital – July 2009

In July 2009 a whois query would reveal five Class C networks (207.23.32.0 - 207.23.36.0) were registered to Surrey Memorial Hospital (NETBLK-SMH-NET1-NET) in Surrey, British Columbia. Today this netblock has been subsumed by its parent, BCNET, a provider of high speed networks to institutions of higher education, research, and health (BCNET, 2009).

Beginning in late July port 80 proxies from this netblock began their debut in published proxy lists. This particular outbreak was over before mid-August, with a handful of proxies appearing again in September and October.

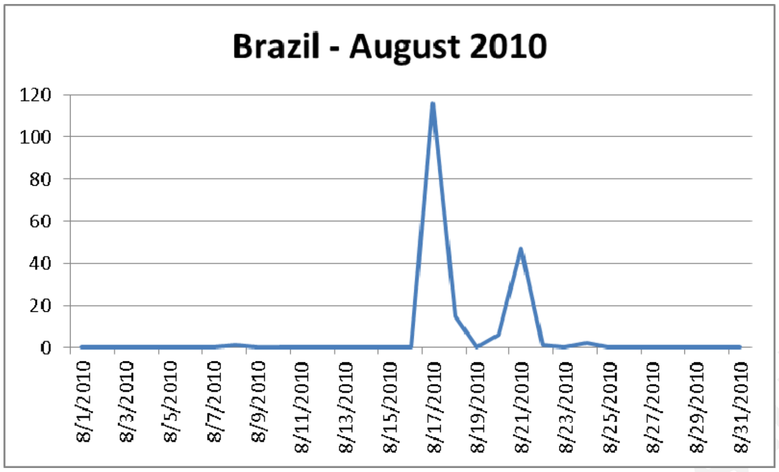


This incident follows the same misconfigured device pattern seen in Macau and Bahrain: confined to a particular address range and displaying the default port (80). Given the disappearance of the SMH netblock and general lag of whois updates it is difficult to say with any certainty that these devices were located in Surrey Memorial Hospital offices at the time of the incident. These addresses were never seen before and have not been seen since, but even with the Koobface outbreaks they represent over two thirds (69%) of all proxies ever identified in Surrey, BC as of December 2010. A random traceroute sampling of the addresses involved in this incident currently return “network unreachable”.

3.6.4. Universidad Unigranrio (Brazil) August 2010

Proxy list users are no strangers to Brazilian proxies. Once considered the hacking capital of the world (Gibb, 2004), Brazil is a proxy gold mine, ranking third behind China and the U.S.A. in the number of active proxies available on any given day.

But hackers or not, Brazilian administrators have been known to make their share of configuration mistakes. In August 2010, administrators of the distance learning program of Universidad Unigranrio opened up the 187.49.166.0/24 subnet to proxy hunters. And there was a proxy at nearly every address.



Although this was a small event, representing only 25% of all *reported* Brazilian proxies in August of 2010, 36% of all *active* Brazilian proxies came from this group.

4. Conclusion

Public proxy lists represent the handiwork of thousands of independent individuals with varying levels of skill. Taken individually, a proxy list site is less than useful, but when aggregated they offer an abundance of information about the condition of the Web at large. With the proper baseline for comparison, an analyst can quickly spot trends in the Internet ecosystem.

This study barely scratches the surface of the harvestable data presented by public proxy lists. Koobface, with its preference for specific ports represents the low hanging fruit. Other malware, such as the Sinowal, Waldec, and BlackEnergy botnets, leveraged random proxy ports for both HTTP and SOCKS proxies. Although SOCKS proxies were not the primary subject of this study, data was inevitably harvested and overwhelming evidence of random port proxies was collected. Even with this element of randomness, trends revealed by date, geolocation, and reverse DNS are easily detected.

The research for this paper was performed with common tools and services available to most US Internet users. A more sophisticated, capable, and properly funded system would likely harvest an order of magnitude more data. Botnet detection systems that rely on sophisticated inline scanners and gigabytes of network flows are worthwhile on a local scale, but globally nothing matches the minable data available in online proxy lists.

5. References

Baltazar, J., Costoya, J., & Flores, R. (2009, July 30). *THE REAL FACE OF KOOBFACE: THE LARGEST WEB 2.0 BOTNET EXPLAINED.* Retrieved August 12, 2010, from Trend Micro: http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/the_real_face_of_koobface_jul2009.pdf

Berners-Lee, T., Fielding, R., & Frystyk, H. (1996, May). *rfc1945.* Retrieved Aug 2010, from The Internet Engineering Task Force (IETF): http://www.ietf.org/rfc/rfc1945.txt

Chien, E., & Shearer, J. (2008, August 3). *W32.Koobface.* Retrieved August 7, 2010, from Symantec.com: http://www.symantec.com/security_response/writeup.jsp?docid=2008-080315-0217-99

Computer Associates. (2006, January 18). *SillyProxy C.* Retrieved December 16, 2010, from Computer Associates: http://arcserve.com/us/securityadvisor/pest/pest-details.aspx?id=453096891#tools

CPJ. (2006, May 2). *10 Most Censored Countries.* Retrieved 12 16, 2010, from Committee to Protect Journalists: http://www.cpj.org/reports/2006/05/10-most-censored-countries.php

Danchev, D. (2008, September 16). *EstDomains and Intercage vs. Cybercrime.* Retrieved 12 16, 2010, from CircleID: http://ddanchev.blogspot.com/2008/09/estdomains-and-intercage-vs-cybercrime.html

Deibert, R., & Rohozinski, R. (2010). *Koobface: Inside a Crimeware Network.* Toronto: Canada Centre for Global Security Studies.

Fortinet. (2010, November 20). *Threat Landscape Report - November 2010 Edition.* Retrieved December 16, 2010, from FortiGuard Center: http://www.fortiguard.com/report/roundup_november_2010.html

itwire. (2009, July 9). *Kaspersky Lab Detects Over 575 New Variants of Koobface in June 2009.* Retrieved December 16, 2010, from itwire.

Kahn, R. (2010, December 3). *Puppy Scams.* Retrieved December 8, 2010, from Court House News Service: http://www.courthousenews.com/2010/12/03/32307.htm

McQuaid, J. (2010, February 6). *Russians Stage Large-scale, Successful Attack on U.S. and State Governments Computers.* Retrieved August 8, 2010, from Secure Home Networks: http://securehomenetwork.blogspot.com/2010/02/russians-stage-large-scale-successful.html

Microsoft. (2009, August 20). *TrojanProxy:Win32/Koobface.gen!F.* Retrieved August 8, 2010, from Microsoft Malware Protection center: http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=TrojanProxy:Win32/Koobface.gen!F

Microsoft. (2009, September 6). *TrojanProxy:Win32/Koobface.gen!Q.* Retrieved August 8, 2010, from Microsoft Malware Protection Center: http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=TrojanProxy%3AWin32%2FKoobface.gen!Q

Munchausen, B. (2010, May 8). *Proxy.org Forum RSS Feed.* Retrieved August 2010, from Proxy.org: http://proxy.org/forum/news-updates/1286-baron-acquires-glype.html

NIPC. (1999, October 22). *ADVISORY 99-024.* Retrieved August 2010, from National Infrastructure Protection Center Press Room: http://www.nipc.gov/warnings/advisories/1999/99-024.htm (defunct)

O'Hagan, J. (2008, September 20). *I got pwned and blacklisted, now what?! .* Retrieved August 8, 2010, from 1 Laptop : 1 Student : http://1laptop1student.blogspot.com/2008/09/i-got-pwned-and-blacklisted-now-what.html

Pai, V., Wang, L., Park, K., Pang, R., & Petersen, L. (2004). The Dark Side of the Web: An Open Proxy's View. *Proceedings of the Second Workshop on Hot Topics in Networking (HotNets-II)* (pp. 2-5). Princeton University.

Robielos, R. (2009, July 2). *Win32/SillyProxy.DG.* Retrieved December 12, 2010, from Computer Associates: http://www.ca.com/us/securityadvisor/virusinfo/virus.aspx?id=79012

Tallahassee Democrat. (2008, March 25). *Secure Florida - News Entry.* Retrieved August 8, 2010, from SecureFlorida.org: http://www.secureflorida.org/news/securityissues/2008/03/25/1323/

Unknown. (2009, June 18). *Podmena, podmena.dll and podmena.sys = spoof, spoof.dll, spoof.sys.* Retrieved September 6, 2009, from Threatfire Research Blog: http://blog.threatfire.com/2009/06/podmena-podmenadll-and-podmenasys-spoof-spoofdll-spoofsys.html

US State Dept. (2010, October 14). *Cameroon.* Retrieved 12 16, 2010, from US Department of State: http://www.state.gov/r/pa/ei/bgn/26431.htm

US-Cert. (2001, April 6). *Vulnerability Note VU#991240.* Retrieved August 2010, from US-Cert: http://www.kb.cert.org/vuls/id/991240

Vamosi, R. (2008, December 4). *Koobface virus hits Facebook.* Retrieved August 8, 2010, from cnet News: http://news.cnet.com/koobface-virus-hits-facebook/

White, D. (2010, August 8). *Social Media Growth from 2006 to 2010.* Retrieved December 16, 2010, from D. Steven White: http://dstevenwhite.com/2010/08/08/social-media-growth-from-2006-to-2010/

Wikipedia. (2002, April 2). *Internet Censorship - Proxy Servers.* Retrieved August 8, 2010, from Wikipedia: http://en.wikipedia.org/wiki/Internet_censorship#Proxy_websites