Detecting Stealth Web Pages That Use Click-Through Cloaking

Strider Search Ranger Report - Part 4

Yi-Min Wang
Ming Ma

December 2006

Technical Report
MSR-TR- 2006-178

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Detecting Stealth Web Pages That Use Click-Through Cloaking

Yi-Min Wang and Ming Ma

Cybersecurity & Systems Management Group, Microsoft Research, Redmond, WA
{ymwang, mingma}@microsoft.com

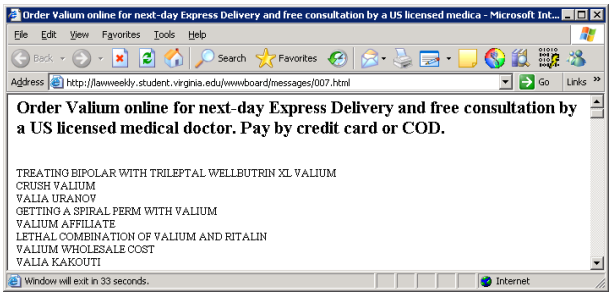
Abstract

Search spam is an attack on search engines' ranking algorithms to promote spam links into top search ranking that they do not deserve. Cloaking is a well-known search spam technique in which spammers serve one page to search-engine crawlers to optimize ranking, but serve a different page to browser users to maximize potential profit. In this experience report, we investigate a different and relatively new type of cloaking, called Click-Through Cloaking, in which spammers serve non-spam content to browsers who visit the URL directly without clicking through search results, in an attempt to evade spam detection by human spam investigators and anti-spam scanners.

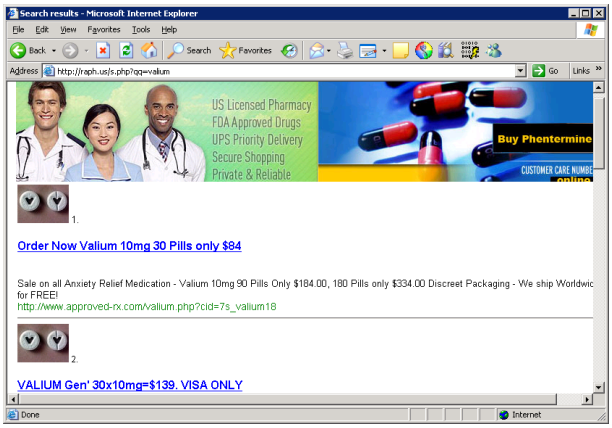
We survey different cloaking techniques actually used in the wild and classify them into three categories: server-side, client-side, and combination. We propose a redirection-diff approach to spam detection by turning spammers' cloaking techniques against themselves. Finally, we present eight case studies in which we used redirection-diff in IP subnet-based spam hunting to defend a major search engine against stealth spam pages that use click-through cloaking.

1. Introduction

Search spammers (or ***web spammers***) refer to those who use questionable search engine optimization techniques to promote their links into top search results. ***Cloaking*** [1,2,3] is one such technique in which the spammers *serve one page to search-engine crawlers to optimize ranking, but serve a different page to browser users to maximize profit.* Figure 1 shows such an example where the spammer gives crawlers a keyword-stuffed page to index (see (a)) but redirect browser users to an *ads-portal page* with numerous drug purchase-related links (see (b)). Such "*crawler-browser cloaking*" behavior can be achieved through "*scripting-on/off cloaking*" in which the same page that contains both scripts and static text is provided to the crawlers (which do not execute scripts and so see the text) as well as to the browsers (which normally execute scripts and so see a rewritten or redirected page).



(a) Keyword-stuffed page indexed by search crawlers



(b) Ads-portal page from the spammer domain raph.us seen by browser users who click through search results

(c) Bogus page seen by anti-spam scanners and human spam investigators who visit the spam URL directly without clicking through a search result

Figure 1: three different pages shown by the same click-through cloaked spam doorway URL lawweekly.student.virginia.edu/wwwboard/messages/007.html in October 2006

In a recent paper [4], we reported a different and relatively new type of cloaking, called ***Click-Through Cloaking***, and presented a preliminary study showing

that a significant percentage of spam blogs created on a major blog site adopted this new approach. Spammers use click-through cloaking to implement stealth web pages by *serving a non-spam page to browser users who visit the URL directly without clicking through a search result*. It is designed to evade spam detection by anti-spam scanners and human spam investigators. For example, by redirecting non-click-through visitors to a bogus non-existent page such as the one shown in (c), the spammers hope to hide their behind-the-scenes, ads-serving domains from spam investigation.

In this report, we provide an in-depth analysis of different techniques for achieving click-through cloaking, and focus on using cloaked pages that have successfully spammed major search engines as seeds to hunt for more spam URLs and eliminate them to improve the quality of search results. In Section 2, we give a brief overview of various behaviors exhibited by spam pages that use click-through cloaking. In Section 3, we give a comprehensive survey of different cloaking techniques, divide them into three categories, and analyze their strength and weaknesses. In Section 4, we give an example of malicious websites that also use cloaking to evade security investigation. We describe the design of our anti-cloaking scanner and redirection-diff spam detection tool in Section 5, and present eight case studies in Section 6 to demonstrate the tool's effectiveness in identifying spam. Section 7 summarizes the paper. All spam pages investigated in this report were active during all or part of the time period between September and November 2006. Since many of them were "throw-away" pages created on free hosting websites as *doorways* to redirect to spammer-operated domains, some of them might have a short lifetime and are no longer active.

2. Behavior of Cloaked Spam Pages

Spammers are in the business to make money. So when users click through search results to reach their pages, they want to show content that has commercial values. Broadly, such content can be divided into three categories: (1) ads-portal pages from which spammers make money by participating in pay-per-click programs; (2) merchant websites which spammers directly own or get paid from through traffic affiliate programs; many casino, pornography, mp3, and travel websites belong to this category; (3) malicious scripts that exploit browser vulnerabilities to install malware programs that steal personal information for illegal purposes. It's not uncommon to see malicious websites simply close the browser window after a successful exploit.

When spam pages encounter non-click-through visitors, the spammers know that they are very likely under investigation; so they want to show non-spam content that minimizes potential damages. We summarize five different cloaking behaviors that we have observed during an extensive, 6-month spam investigation.

(1) "Page not found" message: the spam page pretends to be non-existent and sometimes claims that you must have made a typo.

(2) "Page has been deleted for abuses" (e.g., violations of terms-of-use): this is trying to convince you that somebody else has reported the spam and the problem has been taken care of.

(3) Redirecting to known-good sites such as google.com or msn.com: this attempts to bypass automatic anti-spam scanners that white-list these known-good sites.

(4) Staying on the current page (e.g., a blog page or an empty page): this is to avoid exposing the behind-the-scenes redirection domains.

(5) Redirecting to fake spam-reporting websites: for example, spampatrol.org is a commonly seen redirection target for cloaked spam pages. It asks for your name and email address and promises that *"This site will be closed in five days for a comment and e-mail spam"* (see Figure 2). However, as shown in Case #3 in Section 6, spampatrol.org shares the same IP subnet as many other suspicious drugs- and porn-related websites that use cloaking and is most likely a fake spam-reporting site.

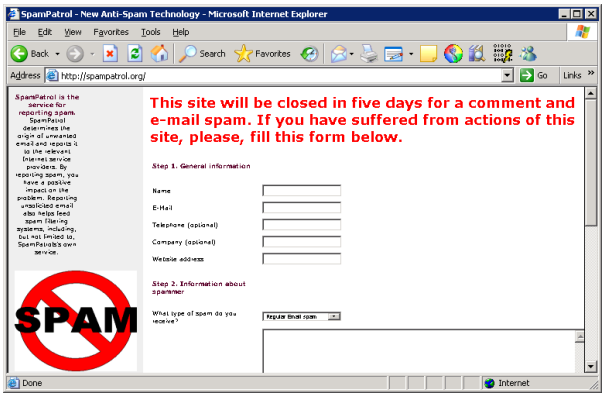


Figure 2: Bogus spam-reporting website that asks for personal information

3. Click-Through Cloaking Techniques

We divide click-through cloaking techniques into three categories: ***server-side cloaking***, ***client-side cloaking***, and ***combination techniques***. We also

distinguish ***simple cloaking***, which only tries to differentiate click-through and non-click-through visitors, from ***advanced cloaking***, which additionally tries to identify click-through visitors who use unusual search strings and are most likely doing spam investigation.

3.1. Server-Side Cloaking

3.1.1. Simple Server-Side Cloaking

The simplest way to achieve click-through cloaking is for web servers to check the Referer field in the header of each incoming HTTP request. If the referrer is a search engine URL, the server assumes that the request came from a search-result click-through and serves the spam content; otherwise, the server returns a bogus page. For example, www.intheribbons.com/win440/2077_durwood.html is a spam URL that uses simple server-side cloaking to serve spam content from lotto.gamblingfoo.com to click-through users but serve a bogus "404 Not Found" page to non-click-through visitors.

Simple server-side cloaking can be easily defeated: a spam investigator could perform a query of "*url:www.intheribbons.com/win440/2077_durwood.html*" at live.com or yahoo.com (or an equivalent "*info:*" query at google.com) to obtain a link to the spam page and click through that link to visit the page. The spammers will be fooled into serving the spam content because the Referer field in the HTTP header is indeed a URL from a major search engine.

3.1.2. Advanced Server-Side Cloaking

Advanced server-side cloaking addresses the weakness by distinguishing spam investigation-style queries from regular search queries. For example, *"url:"* (or *"info:"*), *"link:"*, *"linkdomain:"*, and *"site:"* queries are commonly used by spam investigators, but rarely used by regular users. So a spam server can look for these search strings in the HTTP Referer field and serve cloaked pages.

For example, clicking on acp.edu/phentermine.dhtml from a regular search-result page would return a spam ads-portal page full of drugs-related links, but directly visiting the URL would return a bogus "HTTP 403 (Forbidden)" page. Doing a *"site:www.acp.edu phentermine"* query at live.com and then clicking through the link would still return the bogus page because the spam server sees the *"site:"* query. But issuing a query of *"Order by Noon Est Time, get it tomorrow or choose 2nd day FedEx To All US States"* (where the search string was copied from the page's brief summary displayed in the *"site:"* search-result page) and then clicking on the link would fool the server into serving the spam content.

3.2. Client-Side Cloaking

A major weakness of server-side cloaking, simple or advanced, is that the server cannot tell whether the Referer field in the HTTP header is the "authentic" one generated by the browser, or a fabricated one inserted by an anti-cloaking spam detection program. We have implemented such a program and tested it against spam URLs that use server-side cloaking. We were able to fool all of them into serving spam content by directly visiting them with an inserted Referer field, without clicking through any search results.

This weakness of server-side cloaking and the increasing popularity among spammers to set up throw-away doorway pages on free hosting servers that they do not own motivated the use of client-side cloaking.

3.2.1. Simple Client-Side Cloaking

The basic idea of client-side cloaking is to run a script on the client machine to check the local browser's document.referrer variable. Figure 3 shows an actual script used by the spam URL naha.org/old/tmp/evans-sara-real-fine-place/index.html. It checks if the document.referrer string contains the name of any of the major search engines. If the check succeeds (i.e., the "exit" variable remains true), it redirects the browser to ppcan.info/mp3re.php to continue the redirection chain which eventually leads to spam content; otherwise, it stays on the current doorway page. Since this spam URL does not use advanced cloaking, issuing a query of *"url:http://www.naha.org/old/tmp/evans-sara-real-fine-place/index.html"* at yahoo.com and clicking through the link would reveal the spam content.

More and more spam URLs are using obfuscated scripts to perform client-side cloaking in order to evade content-based detection by crawlers and human spam investigators. Figure 4 shows a sample obfuscated script fragment used by the spam URL buyviagralive.blogspot.com. By replacing document.write() with alert(), we were able to de-obfuscate the script and see the cloaking logic that performs a similar check of document.referrer against major search engines' names as well as their specific URL structures.

```
var url = document.location + ""; exit=true;
ref=escape(document.referrer);
if ((ref.indexOf('search')==-1) && (ref.indexOf('google')==-1)
&& (ref.indexOf('find')==-1) && (ref.indexOf('yahoo')==-1) &&
(ref.indexOf('aol')==-1) && (ref.indexOf('msn')==-1) &&
(ref.indexOf('altavista')==-1) && (ref.indexOf('ask')==-1) &&
(ref.indexOf('alltheweb')==-1) && (ref.indexOf('dogpile')==-1)
&& (ref.indexOf('excite')==-1) && (ref.indexOf('netscape')==-1)
&& (ref.indexOf('fast')==-1) && (ref.indexOf('seek')==-1) &&
```

```
(ref.indexOf('find')==-1) && (ref.indexOf('searchfeed')==-1) &&
(ref.indexOf('about.com')==-1) && (ref.indexOf('dmoz')==-1)
&& (ref.indexOf('accoona')==-1) && (ref.indexOf('crawler')==-
1)) { exit=false; } if (exit) { p=location;
r=escape(document.referrer);
location='http://ppcan.info/mp3re.php?niche=Evans,
Sara&ref='+r }
```

Figure 3: A basic client-side cloaking script

```
<script
language="javascript">document.write("\x3c\x73\x63\x72\x69\
x70\x74\x3e\x20\x76\x61\x72\x20\x72\x3d\x64\x6f\x63\x75\x6
d\x65\x6e\x74\x2e\x72\x65\x66\x65\x72\x72\x65\x72\x2c\x7
4
...
x6e\x2e\x70\x68\x70\x3f\x72\x3d" + "blogspot" +
"\x26\x67\x3d" + "pharmacy" + "\x26\x6b\x3d" + "Buy Viagra"
+
"\x22\x3b\x20\x3c\x2f\x73\x63\x72\x69\x70\x74\x3e");</script>
```

Figure 4: Obfuscated script (the encoded string in bold face translates into "document.referrer")

3.2.2. Advanced Client-Side Cloaking

Like advanced server-side cloaking described in Section 3.1.2, many client-side cloaking pages perform advanced checks, as shown in Figure 5 for lossovernigh180.blogspot.com. In addition to checking for *"link:"*, *"linkdomain:"*, and *"site:"*, it also performs a general check of whether the spam URL's domain name appears as part of the referrer string, which covers the cases of *"url:"* and *"info:"* queries. The result of this check decides the output of the is_SE_traffic() function, based on which either a spam page or a bogus non-existent page is served.

```
Function is_se_traffic() {
  if ( document.referrer ) {
    if ( document.referrer.indexOf("google")>0
      || document.referrer.indexOf("yahoo")>0
      || document.referrer.indexOf("msn")>0
      || document.referrer.indexOf("live")>0
      || document.referrer.indexOf("search.blogger.com")>0
      || document.referrer.indexOf("www.ask.com")>0)
    {
      If ( document.referrer.indexOf( document.domain )<0
          && document.referrer.indexOf( "link%3A" )<0
          && document.referrer.indexOf( "linkdomain%3A" )<0
          && document.referrer.indexOf( "site%3A" )<0 )
      { return true; }
    }
  }
  return false;
}
```

Figure 5: Script fragment for document.referrer checking from zeppele.com/9726_5fcb7_Vp8.js, which lossovernigh180.blogspot.com redirects to

3.3. Combining Client-Side Script with Server-Side Checking

A major weakness of client-side cloaking techniques, simple or advanced, is that the cloaking logic is exposed to spam investigators, who can then design the most effective anti-cloaking tool or procedure accordingly. To get the best of both worlds (i.e., extracting referrer information directly from the client-side document.referrer variable and hiding the cloaking logic on the server side), some spammers have migrated to a combination cloaking solution.

The following spam URL hosted on a university web site used combo cloaking lawweekly.student.virginia.edu/wwwboard/messages/007.html: it uses a client-side script to extract the document.referrer information and *reports it to the spam domain 4nrop.com* as part of the URL. If the referrer information passes the server-side check, the browser is redirected to a spam page hosted on raph.us; otherwise, it is redirected to lawweekly.student.virginia.edu/404.html, which is a bogus non-existent page. This spammer has attacked several other .edu websites and set up cloaked pages with similar behavior; pbl.cc.gatech.edu/bmed3200a/10 and languages.uconn.edu/faculty/CVs/data-10.php are just two such examples

Figure 6 shows an example of obfuscated combo cloaking used by the spam URL mywebpage.netscape.com/superphrm2/order-tramadol.htm. The script feeds an obfuscated string to the obfuscated function kqqw() to generate another script code to be executed by eval(). By replacing eval() with alert(), we were able to see that the script eventually reports the document.referrer information to the spam server emaxrdr.com, which then redirects the browser either to a spam page hosted on pillserch.com or to the suspicious website spampatrol.org described previously.

```
<script> var params="f=pharmacy&cat=tramadol";
function kqqw(s){
  var Tqqe=String("qwertyuioplkjhgfdsazxcvbnmQWERTYU
  IOPLKJHGFDSAZXCVBNM_1234567890");
  var tqqr=String(s); var Bqqt=String("");
  var Iqqy,pqqu,Yqqi=tqqr.length;
  for ( Iqqy=0; Iqqy<Yqqi; Iqqy+=2) {
    pqqu=Tqqe.indexOf(tqqr.charAt(Iqqy))*63;
    pqqu+=Tqqe.indexOf(tqqr.charAt(Iqqy+1));
    Bqqt=Bqqt+String.fromCharCode(pqqu);
  }
  return(Bqqt);
}
eval(kqqw('wKwVwLw2wXwJwCw1qXw4wMwDw1wJqGqHq8
qHqSqHw_ ...
Bw1qHqSqHq0qHqFq7'));</script>
```

Figure 6: Obfuscated script for combo cloaking

4. Cloaked Malicious Web Pages

We previously developed the Strider HoneyMonkey system for detecting malicious websites that attempt to exploit browser vulnerabilities [5]. These HoneyMonkeys, running inside unpatched Virtual

Machines (VMs), mimic human browsing activities by launching an actual browser to visit each suspect website, and later identify malicious ones by detecting drive-by software installation outside browser sandbox. The system has successfully detected thousands of websites that were exploiting known and zero-day vulnerabilities.

In our recent search spam investigation, we have found thousands of spam pages that are malicious. That is, some malicious website operators are using search spamming techniques to push their URLs into top search results in major search engines in order to draw more traffic to exploit. We have also discovered hundreds of malicious spam pages that used click-through cloaking. For example, on October 13, 2006, the malicious URL mandevillechevrolet . com (spaces added for safety) appeared as the #1 Yahoo search result for *"mandeville chevrolet"*, which would install a malware program named "ane.exe" under C:\ if the clicking user's machine is vulnerable. To avoid being detected by systems like HoneyMonkey, the spam page used a client-side script to check document.referrer and only redirected the browser to the malicious porn site hqualityporn . com if the visit came from a search click-through. This demonstrates the importance for exploit detection systems and human security investigators to use anti-cloaking techniques to be discussed next.

5. Anti-Cloaking Scanner and Redirection-Diff Spam Detection Tool

To effectively detect all spam pages that use click-through cloaking, we adopt an end-to-end approach of building an anti-cloaking scanner that always visits websites by clicking through search-result pages, instead of trying to exploit the weakness of individual cloaking techniques. Given a suspect URL, the scanner *derives from the URL name the likely keywords that the spammer is targeting*, *queries Live Search to obtain a search-result page that correctly sets the document.referrer variable*, *inserts a link to the suspect URL into the page*, and *generate a click on that link*. The scanner also incorporates the Strider URL Tracer [6] to record all *third-party redirection domains* reached as a result of the visit. If a redirection domain belongs to a known spammer, the URL is flagged as spam; otherwise, further investigation to gather evidence of spam activities is required.

We found that the use of click-through cloaking is almost always an indication of spam because good websites do not engage in such deceptive behavior. Therefore, we propose a ***redirection-diff*** approach to turn spammers' cloaking techniques against themselves and use it as a detection mechanism [7]. Specifically, we *scan each suspect URL twice – one with anti-cloaking and one without, compare the two vectors of redirection domains, and flag those that exhibit a difference in the comparison*.

Table 1 shows that redirection-diff can detect all eight cloaked URLs discussed so far with no false negatives. However, in practice, there are several possibilities for false positives: (1) some websites serve rotating ads from different domains. (2) Some websites rotate final destinations to distribute traffic among multiple downstream sites. (We use the term "***final destination***" to refer to *the URL in the address bar when all redirections have been finished*.) (3) Some web accesses may fail due to transient network or server problems. Although spam judgment is ultimately a subjective matter that requires humans to make the final determination, it is important for the tool to have a low false-positive rate in order to minimize expensive manual effort.

Table 1: Redirection-Diff – spam URL: [vector with anti-cloaking] vs. [vector without anti-cloaking] with diff highlighted in bold face

1. lawweekly.student.virginia.edu/wwwboard/messages/007.html: [4nrop.com, **raph.us, 8-d.com**] vs. [4nrop.com]
2. www.intheribbons.com/win440/2077_durwood.html: [**gamblingfoo.com**] vs. [none]
3. www.acp.edu/phentermine.dhtml: [**searchfeed.com**] vs. [none]
4. www.naha.org/old/tmp/evans-sara-real-fine-place/index.html: [ppcan.info, **mp3sugar.com**] vs. [ppcan.info]
5. buyviagralive.blogspot.com/: [blogger.com, **trafficmanager.info, 4yousearch.com**] vs. [blogger.com]
6. lossovernigh180.blogspot.com/: [zeppele.com, **worlddatinghere.com, adultfriendfinder.com**] vs. [zeppele.com, **follar-sexo-conocer.com**]
7. mywebpage.netscape.com/superphrm2/order-tramadol.htm: [aol.com, atwola.com, doubleclick.net, advertising.com, adsdk.com, emaxrdr.com, **pillserch.com**] vs. [aol.com, atwola.com, doubleclick.net, advertising.com, adsdk.com, emaxrdr.com, **spampatrol.org, statcounter.com**]
8. www . mandevillechevrolet . com /: [**hqualityporn.com, dinet.info, frlynx.info, joutweb.net**] vs. [none]

In practice, we found that the following modifications to the basic redirection-diff approach are effective in reducing false positives: (1) the majority of today's cloaked pages can be detected by comparing only the final destinations from the two scans (i.e., redirection-diff of size-1 vectors). (2) For each suspect URL, we can perform the scans and diff multiple times and exclude those that do not result in a consistent diff result (see Case #1 in Section 6). (3) Given a group of

URLs that are expected to exhibit a similar cloaking behavior, we can perform the scans and diff for each URL just once and exclude those that are not consistent with the rest of the group. For example, given a confirmed cloaked URL, we may construct a group of suspect URLs by examining domains hosted on nearby IP addresses [8] (see Cases #2~#6). We show in the next section that we have used this cloaked-spam hunting technqiue to successfully identify over 10,000 spam pages and hundreds of new spammer redirection domains.

6. Case Studies of Click-Through Cloaking

Case Study #1: False-positive Evaluation

In one experiment, human experts were given the top-10 results of a search benchmark, consisting of hundreds of keywords, from a major search engine, and they identified 736 URLs as spam. Since our goal was to establish a lower bound on the percentage of cloaked spam URLs, we compared only the final destinations in redirection-diff. The first scan of these 736 URLs flagged 53 of them as suspicious. The second scan excluded two of the 53 that had inconsistent diff results (both due to rotating final destinations). We then manually examined the remaining 51 and found that only one (***2%***) of them was a false positive, again due to rotating final destinations. The lower-bound cloaking percentage among spam URLs was therefore ***50/736=6.8%*** for this benchmark.

Case Study #2 ~ #6: IP Subnet-based Spam Hunting

Table 2 shows the results from Case Study #2 ~ #6, in which we performed IP address-based spam hunting by starting with a seed URL that was successfully promoted into top search results (shown in the top row of each case). The second column shows the range of suspect IP addresses surrounding the one that hosted the seed URL. The third column shows the number of domains hosted on those IP addresses as well as the number of URLs we obtained for scanning by issuing a *"site:"* query for each domain.

The fourth column shows the number of cloaked URLs detected by our tool based on comparing final destinations only, and the fifth column shows the number of final destinations that were hiding through cloaking. In summary, we found that IP-based spam hunting was very effective, identifying ***33-99%*** of suspicious URLs as cloaked URLs in these five cases. In total, we discovered ***11,973*** unique cloaked URLs associated with ***241*** unique hiding final-destination domains, many of which were previously unknown spammer redirection domains.

Table 2: IP Subnet-based Spam Hunting Results

	Suspicious IP address range	# domains / # scanned URLs	# cloaked URLs	# hiding final destinations
#2	moped-scooter.ngvjj.info/moped-motor-scooter.html			
	217.11.233.224 ~254	325 / 8,444	6,926 ***(82%)***	110
#3	spampatrol.org			
	67.19.92.170 ~174	70 / 375	292 ***(78%)***	4
#4	www.intheribbons.com/win440/2077_durwood.html			
	69.59.158.96 ~111	17 / 750	743 ***(99%)***	21
#5	mobile.qode.info/arabic-mp3-ringtone.html			
	81.0.195.190 ~205	233 / 5,306	2,459 ***(46%)***	125
#6	samsung.yourphoneonline.net/movie_fone			
	66.29.15.128 ~135	94 / 4,649	1,553 ***(33%)***	2

Case #3 is unique in that the seed URL is not a cloaked URL, but rather a suspicious final destination for cloaked pages, as discussed previously. Our scan results show that a large percentage of sampled URLs hosted on its nearby IP addresses are cloaked URLs and they all share spampatrol.org as their bogus final-destination page, which clearly indicates that this is a fake spam-reporting page. (Interestingly, we discovered another similar spam-reporting site abusepost.com in a separate scan of cloaked URLs including hometown.aol.com/ftvgirls55/ftv.html.)

During the investigation of Case #6, we encountered an interesting false-positive issue due to *self-clicking ads-portal pages*. Some of the suspect URLs redirected to allishere.us/in.php?id=404 which was a final-destination, ads-portal page that would non-deterministically and automatically generate a click on one of the ads links if left unattended. That would change the recorded final-destination domain and introduce false diff results. Fortunately, we had thousands of suspect pages that followed the "template" cloaking behavior of the seed URL and generated the two vectors as [<one of two final destinations>] vs. [none]. So it was fairly easy to simply (conservatively) exclude those diff results caused by the random clicks.

Case Study #7: Redirection-diff of Full Vectors

Taking this cloaked URL from Section 3.1.2 as a seed: www.acp.edu/phentermine.dhtml, we issued a *"site:www.acp.edu"* query to retrieve the top-1000 results and extracted 825 .dhtml URLs. Since this set of URLs do not land on a third-party final destination in either scan, they require a diff of the full redirection vectors. In the first pass of the analysis, our tool detected 554 (***67%***) of the 825 URLs as cloaked pages through the diff of [searchfeed.com] vs. [none]; the

remaining 33% did not generate any third-party traffic in either scan. Based on the observation that actual spam pages all fetched images from "www.acp.edu/images/" but none of the cloaked bogus pages did, we extended the redirection vector to include this non-third-party URL prefix and were able to confirm that all 825 URLs used cloaking.

In fact, we later discovered that this site was actually hosting a keyword-based ads engine that could generate an infinite number of clocked URLs. For example, visiting this arbitrarily constructed URL www.acp.edu/garbage-in-garbage-out.dhtml with our anti-cloaking scanner would return a list of ads based on the keywords *"garbage in garbage out"*, while visiting the URL directly would return a bogus page.

Case Study #8: Malicious URL Spam Hunting

Taking this malicious URL from Section 4 as a seed: mandevillechevrolet . com, we extracted 118 suspicious domains hosted on its nearby IP addresses. Since the cloaked malicious behavior was exhibited at the domain level, we scanned only the 118 domain-level pages. Our tool detected 90 (***76%***) of the 118 URLs as cloaked URLs that were hiding these three behind-the-scenes malicious domains: dinet.info, frlynx.info, and joutweb.net. When we re-scanned this group of URLs in mid-November 2006, these three domains were replaced by tisall.info, frsets.info, and recdir.org (hosted on the same pair of IP addresses 85.255.115.227 and 66.230.138.194), while the cloaking behavior remained the same.

7. Summary

Search engines have become such a dominating web portal that many spammers are willing to sacrifice all non-click-through traffic by using click-through cloaking in order to minimize the risk of getting caught and blacklisted. We have provided an in-depth analysis of stealth spam pages that use referrer-based cloaking to hide spammer-operated redirection domains from spam investigation. We have categorized cloaked pages at three levels: at the first level, we differentiate server-side cloaking, client-side cloaking, and combination techniques; at the second level, we distinguish advanced cloaking, which checks for spam investigation-style queries, from simple cloaking; at the third level, we distinguish obfuscated scripts from plaintext referrer-checking scripts.

We have implemented an anti-cloaking scanner that always visits websites by clicking through search results in order to defeat all referrer-based cloaking techniques. We have also implemented a redirection-diff tool that turns the cloaking behavior into a spam detection mechanism. Through IP subnet-based spam hunting, we have been able to use the tool to discover over 10,000 cloaked pages that were hiding hundreds of spam-related redirection domains – a level of prevalence that came as a surprise to us. We have also shown that malicious website operators are using cloaking techniques as well, so it is important for automated exploit detection systems and human security investigators to adopt anti-cloaking techniques in their scanning and investigation.

References

[1] Gyongyi, Z. and Garcia-Molina, H., "Web Spam Taxonomy," in *Proc. International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[2] Wu, B. and Davidson, B.D., "Cloaking and Redirection: A Preliminary Study," in *Proc. AIRWeb,* 2005

[3] Chellapilla, K. and Chickering, D.M., "Improving Cloaking Detection Using Search Query Popularity and Monetizability," in *Proc. AIRWeb,* August 2006

[4] Niu, Y., Wang, Y. M., Chen, H., Ma, M., and Hsu, F., "A Quantitative Study of Forum Spamming Using Context-based Analysis," to appear in *Proc. Network and Distributed System Security (NDSS) Symposium*, 2007 (MSR-TR-2006-173)

[5] Wang, Y. M., Beck, D., Jiang, X., Roussev, R., Verbowski, C., Chen, S., and King, S., "Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities," in *Proc. NDSS*, February 2006.

[6] Wang, Y. M., Beck, D., Wang, J., Verbowski, C., and Daniels, B., "Strider Typo-Patrol: Discovery and Analysis of Systematic Typo-Squatting," in *Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, July 2006

[7] Wang, Y. M., Beck, D., Vo, B., Roussev, R., and Verbowski, C., "Detecting Stealth Software with Strider GhostBuster," in *Proc. IEEE International Conference on Dependable Systems and Networks (DSN)*, June 2005.

[8] Reverse IP lookup, http://www.domaintools.com/reverse-ip/.