

Protection from screen scraping

[closed]

Asked 15 years, 11 months ago Modified 12 years, 7 months ago

Viewed 24k times



31



Closed. This question is [off-topic](#). It is not currently accepting answers.



Want to improve this question? [Update the question](#) so it's [on-topic](#) for Stack Overflow.

Closed 12 years ago.

[Improve this question](#)

Following on from my question on the [Legalities of screen scraping](#), even if it's illegal people will still try, so:

What technical mechanisms can be employed to *prevent* or at least disincentivise screen scraping?

Oh and just for grins and to make life difficult, it may well be nice to retain access for search engines. I may well be playing devil's advocate here but there is a serious underlying point.

screen-scraping

Share Follow

edited May 23, 2017 at 11:45



Community Bot

1 • 1

asked Dec 28, 2008 at 22:59



Mat

86.3k • 35 • 94 • 111

See stackoverflow.com/questions/261638/how-do-i-protect-python-code. It's the same question with the language changed from Python to HTML. – [S.Lott](#) Dec 28, 2008 at 23:10

19 The delivery of HTML pages and Python source code are so wildly different that calling this question a duplicate is laughable. – [Bill the Lizard](#) Dec 28, 2008 at 23:12

I also came across 'How to protect/monitor your site from crawling by malicious user' (stackoverflow.com/questions/385069/...). This is not really a dupe but does address similar issues. – [Mat](#) Dec 30, 2008 at 22:17

9 Viewing a web page is just a slow, manual form of screen scraping. – [Instance Hunter](#) Jul 12, 2009 at 4:17

just like DRM this is really a big waste of resources that can be easily overcome in almost every case, especially if the data is worth spending the time to work around whatever you come up with. – [user177800](#) Jan 15, 2010 at 19:09

21 Answers

Sorted by:

Highest score (default)





You can't prevent it.

62

Share Follow

answered Dec 28, 2008 at 23:04



Bombe

83.7k ● 20 ● 127 ● 127



1 I love those answers... "You can't",.. Everything can be done. In one way or another. – [Stefan](#) Dec 28, 2008 at 23:57

7 Ok, you can do it. Just don't output anything. Show your user a blank page. Missing accomplished: screen scraping prevented! – [Bob Somers](#) Dec 29, 2008 at 0:00

"You can't"... "Everything can be done". Two absolutes that are never true. – [Ed Swangren](#) Dec 29, 2008 at 2:00

7 "never true", another absolute... :) – [Bill the Lizard](#) Dec 29, 2008 at 2:43

27 You have to give the user the data (so they can use your page). You have to not give the user the data (or they can scrape it). If you have further problems, consult a Zen master, 'cause this software guy is out of ideas. – [David Thornley](#) Jan 19, 2009 at 22:25



22

I've written a blog post about this here: <http://blog.screen-scraper.com/2009/08/17/further-thoughts-on-hindering-screen-scraping/>

To paraphrase:



If you post information on the internet someone can get it, it's just a matter of how many resources they want to invest. Some means to make the required resources higher are:

Turing tests

The most common implementation of the Turing Test is the old CAPTCHA that tries to ensure a human reads the text in an image, and feeds it into a form.

We have found a large number of sites that implement a very weak CAPTCHA that takes only a few minutes to get around. On the other hand, there are some very good implementations of Turing Tests that we would opt not to deal with given the choice, but a sophisticated OCR can sometimes overcome those, or many bulletin board spammers have some clever tricks to get past these.

Data as images

Sometimes you know which parts of your data are valuable. In that case it becomes reasonable to replace such text with an image. As with the Turing Test, there is OCR software that can read it, and there's no reason we can't save the image and have someone read it later.

Often times, however, listing data as an image without a text alternate is in violation of the Americans with Disabilities Act (ADA), and can be overcome with a couple of phone calls to a company's legal department.

Code obfuscation

Using something like a JavaScript function to show data on the page though it's not anywhere in the HTML source is a good trick. Other examples include putting prolific, extraneous comments through the page or having an interactive page that orders things in an unpredictable way (and the example I think of used CSS to make the display the same no matter the arrangement of the code.)

CSS Sprites

Recently we've encountered some instances where a page has one images containing numbers and letters, and used CSS to display only the characters they desired. This is in effect a combination of the previous 2 methods. First we have to get that master-image and read what characters are there, then we'd need to read the CSS in the site and determine to what character each tag was pointing.

While this is very clever, I suspect this too would run afoul the ADA, though I've not tested that yet.

Limit search results

Most of the data we want to get at is behind some sort of form. Some are easy, and submitting a blank form will yield all of the results. Some need an asterisk or percent put in the form. The hardest ones are those that will give you only so many results per query. Sometimes we just make a loop that will submit the letters of the alphabet to

the form, but if that's too general, we must make a loop to submit all combination of 2 or 3 letters—that's 17,576 page requests.

IP Filtering

On occasion, a diligent webmaster will notice a large number of page requests coming from a particular IP address, and block requests from that domain. There are a number of methods to pass requests through alternate domains, however, so this method isn't generally very effective.

Site Tinkering

Scraping always keys off of certain things in the HTML. Some sites have the resources to constantly tweak their HTML so that any scrapes are constantly out of date. Therefore it becomes cost ineffective to continually update the scrape for the constantly changing conditions.

Share Follow

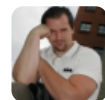
edited Jun 28, 2011 at 20:44



jm.

23.7k ● 23 ● 81 ● 93

answered Aug 27, 2009 at 17:11



Jason Bellows

329 ● 2 ● 7

8 I think you meant OCR (optical character recognition) not ORC. Too much World of Warcraft, dude! – T.Rob Feb 7, 2011 at 5:18



20



So, one approach would be to obfuscate the code (rot13, or something), and then have some javascript in the page that do something like

`document.write(unobfuscate(obfuscated_page))`. But this totally blows away search engines (probably!).

Of course this doesn't actually stop someone who wants to steal your data either, but it does make it harder.

Once the client has the data it is pretty much game over, so you need to look at something on the server side.

Given that search engines are basically screen scrapers things are difficult. You need to look at what the difference between the *good* screen scrapers and the *bad* screen scrapers are. And of course, you have just the normal human users as well. So this comes down to a problem of how can you on the server effectively classify a request as coming from a *human*, a *good* screen scraper, or a *bad* screen scraper.

So, the place to start would be looking at your log-files and seeing if there is some pattern that allows you to effectively classify requests, and then on determining the pattern see if there is some way that a *bad* screen scraper, upon knowing this classification, could cloak itself to appear like a *human* or *good* screen scraper.

Some ideas:

- You may be able to determine the *good* screen scrapers by IP address(es)..
- You could potentially determine scraper vs. human by number of concurrent connections, total number of connections per time-period, access pattern, etc.

Obviously these aren't ideal or fool-proof. Another tactic is to determine what measures can you take that are unobtrusive to humans, but (*may be*) annoying for scrapers. An example might be slowing down the number of requests. (Depends on the time criticality of the request. If they are scraping in real-time, this would effect their end users).

The other aspect is to look at serving these users better. Clearly they are scraping because they want the data. If you provide them an easy way in which to directly obtain the data in a useful format then that will be easier for them to do instead of screen scraping. If there is an easy way then access to the data can be regulated. E.g: give requesters a unique key, and then limit the number of requests per key to avoid overload on the server, or charge per 1000 requests, etc.

Of course there are still people who will want to rip you off, and then there are probably other ways to disincentivise, but they probably start being non-technical, and require legal avenues to be pursued.



benno

2,147 ● 1 ● 20 ● 24

this "solution" doesn't prevent "screen scraping" in any way, I can just save the rendered HTML to a disk and parse it there as much as I like. – user177800 Jan 15, 2010 at 19:03

- 2 I think I addressed very clearly in the answer: "Of course this doesn't actually stop someone who wants to steal your data either". – benno Jan 16, 2010 at 22:50
-



11



It's pretty hard to prevent screen scraping but if you really, really wanted to you could change your HTML frequently or change the HTML tag names frequently. Most screen scrapers work by using string comparisons with tag names, or regular expressions searching for particular strings etc. If you are changing the underlying HTML it will make them need to change their software.

Share Follow

answered Dec 28, 2008 at 23:54



James Sugrue

15k ● 10 ● 62 ● 93

how do propose "changing" standard HTML tags and having browsers display the HTML? This doesn't make any sense – user177800 Jan 15, 2010 at 19:05

- 1 No not what I meant - by changing the HTML, I meant changing the HTML code or structure, as the scraping apps usually expect the HTML code to be in a particular form with particular names. Changing them regularly would mean the scraping app would also need to be recoded. Like I said, not

going to prevent it, but could annoy the scraper sufficiently enough that they stop – [JamesSugrue](#) Jan 18, 2010 at 3:15

- 4 Shiiii don't give good advise on preventing screen scraping... como on.. don't make my job dificult ;) +1 when a page changes too much that's when I evaluate if I still need their data. – [Diego Castro](#) Dec 1, 2010 at 16:02 ✎
-

I think that in a lot of cases, this strategy might be one of the best. The "good" scrapers like search engines will be able to continue categorizing the data, and the "bad" ones, i.e. ones that are written by people just trying to snatch data off of a site will be force to refactor their scraper code. Also, my idea was to create a few difference dynamically shifting HTML structures for pages and/or modules. That way the scrapers would always have to be on their toes.

– [Grizzly Peak Software](#) Nov 18, 2013 at 18:52



5



It would be very difficult to prevent. The problem is that Web pages are *meant* to be parsed by a program (your browser), so they are exceptionally easy to scrape. The best you can do is be vigilant, and if you find that your site is being scraped, block the IP of the offending program.



Share Follow

answered Dec 28, 2008 at 23:07



[Bill the Lizard](#)

405k ● 211 ● 572 ● 889



4

Don't prevent it, detect it and retaliate those who try.

For example, leave your site open to download but disseminate some links that no sane user would follow. If



someone follows that link, is clicking too fast for a human or other suspicious behaviour, react promptly to stop the user from trying. If there is a login system, block the user and contact him regarding unacceptable behaviour. That should make sure they don't try again. If there is no login system, instead of actual pages, return a big warning with fake links to the same warning.

This really applies for things like Safari Bookshelf where a user copy-pasting a piece of code or a chapter to mail a colleague is fine while a full download of book is not acceptable. I'm quite sure that they detect when some tries to download their books, block the account and show the culprit that he might get in REAL trouble should he try that again.

To make a non-IT analogy, if airport security only made it hard to bring weapons on board of planes, terrorists would try many ways to sneak one past security. But the fact that just trying will get you in deep trouble make it so that nobody is going to try and find the ways to sneak one. The risk of getting caught and punished is too high. Just do the same. If possible.

Share Follow

answered Aug 21, 2009 at 9:43



Eric Darchis

26.6k ● 4 ● 29 ● 49



Search engines ARE screen scrapers by definition. So most things you do to make it harder to screen scrape will also make it harder to index your content.

4



Well behaved robots will honour your robots.txt file. You could also block the IP of known offenders or add obfuscating HTML tags into your content when it's not sent to a known good robot. It's a losing battle though. I recommend the litigation route for known offenders.

You could also hide identifying data in the content to make it easier to track down offenders. Encyclopaedias have been known to add [Fictitious entries](#) to help detect and prosecute copyright infringers.

Share Follow

edited Jun 28, 2011 at 20:47



jm.

23.7k ● 23 ● 81 ● 93

answered Dec 28, 2008 at 23:48



Chris Nava

6,802 ● 3 ● 27 ● 31



3



Prevent? -- impossible, but you can make it harder.

Disincentivise? -- possible, but you won't like the answer: provide bulk data exports for interested parties.

On the long run, all your competitors will have the same data if you publish it, so you need other means of diversifying your website (e.g. update it more frequently, make it faster or easier to use). Nowadays even Google is using scraped information like user reviews, what do you think you can do about it? Sue them and get booted from their index?



mjoy

2,767 ● 3 ● 20 ● 22

-1 for using 'Disincentivise' - eschew obfuscation! – [egrinin](#)

Apr 17, 2010 at 19:48

3 it's in the question, not my fault ... – [mjoy](#) Apr 22, 2010 at 14:29



3



The best return on investment is probably to add random newlines and multiple spaces, since most screen scrapers work from the HTML as text rather than as a XML (since most pages don't parse as valid XML).



The browser ignores whitespace, so your user's don't notice that



```
Price : 1
Price :    2
Price\n:\n3
```


are different. (this comes from my experience scraping government sites with AWK).

Next step is adding tags around random elements to mess up the DOM.



Dave

917 ● 1 ● 8 ● 20

Those are easy to get around. This is a weak solution. Changing the HTML would be better like from `` to `< a href = 'link'>` for example. But it is still fairly simple to accomodate for in the code. Yeah it may delay someone for a few minutes, but it can be overcome quickly. – [Chuck Burgess](#) Sep 3, 2010 at 13:55 

- 2 And if the screen scraper is written by a decent programmer, they're using an HTML parser anyway and scraping from the DOM tree, not the HTML source. – [Mark E. Haase](#) Apr 27, 2012 at 19:24
-



2



One way is to create an function that takes text and position and then Serverside generate x, y pos for every character in the text, generate divs in random order containing the characters. Generate a javascript that then position every div on right place on screen. Looks good on screen but in code behind there is no real order to fetch the text if you dont go throuh the trouble to scrape via your javascript (that can be changed dynamically every request)

Too much work and have possibly many quirks, it depends on how much text and how complicate UI you have on the site and other things.

Share Follow

[edited Dec 29, 2008 at 21:35](#)

[answered Dec 28, 2008 at 23:55](#)



Stefan

11.5k ● 8 ● 52 ● 78



1



Very few I think given the intention of any site is to publish (i.e. to make public) information.

- You can hide your data behind logins of course, but that's a very situational solution.
- I've seen apps which would only serve up content where the request headers indicated a web browser (rather than say anonymous or "jakarta") but that's easy to spoof and you'll lose some genuine humans.
- Then there's the possibility that you accept some scrapage but make life insurmountably hard for them by not serving content if requests are coming from the same IP at too high a rate. This suffers from not being full coverage but more importantly there is the "AOL problem" that an IP can cover many *many* unique human users.

Both of the last two techniques also depend heavily on having traffic intercepting technology which is an inevitable performance and/or financial outlay.

Share Follow

answered Dec 28, 2008 at 23:07



annakata

75.7k ● 18 ● 115 ● 180

How will you lose genuine humans if you block request headers that have wget, or curl? – [Henley Wing Chiu](#) Apr 2, 2011 at 5:39



1

Given that most sites want a good search engine ranking, and search engines are scraper bots, there's not much you can do that won't harm your SEO.



You could make an entirely ajax loaded site or flash based site, which would make it harder for bots, or hide everything behind a login, which would make it harder still, but either of these approaches is going to hurt your search rankings and possibly annoy your users, and if someone really wants it, they'll find a way.

The only guaranteed way of having content that can't be scraped is to not publish it on the web. The nature of the web is such that when you put it out there, it's out there.

Share Follow

answered Dec 28, 2008 at 23:19



seanb

6,954 ● 2 ● 34 ● 34



1

If its not much information you want to protect you can convert it to a picture on the fly. Then they must use OCR wich makes it easier to scrape another site instead of yours..



Share Follow

answered Dec 28, 2008 at 23:59



Stefan

11.5k ● 8 ● 52 ● 78



0



You could check the user agent of clients coming to your site. Some third party screen scraping programs have their own user agent so you could block that. Good screen scrapers however spoof their user agent so you won't be able to detect it. Be careful if you do try to block anyone because you don't want to block a legitimate user :)

The best you can hope for is to block people using screen scrapers that aren't smart enough to change their user agent.

Share Follow

answered Dec 28, 2008 at 23:08



Alex

36.4k ● 11 ● 56 ● 68



0



I tried to "screen scrape" some PDF files once, only to find that they'd actually put the characters in the PDF in semi-random order. I guess the PDF format allows you to specify a location for each block of text, and they'd used very small blocks (smaller than a word). I suspect that the PDFs in question weren't trying to prevent screen scraping so much as they were doing something weird with their render engine.

I wonder if you could do something like that.

Share Follow

answered Dec 28, 2008 at 23:12



Paul Tomblin

183k ● 59 ● 323 ● 410



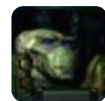
0



You could put everything in flash, but in most cases that would annoy many legitimate users, myself included. It can work for some information such as stock prices or graphs.

Share Follow

answered Dec 28, 2008 at 23:13



too much php

90.8k ● 36 ● 131 ● 139



Google already indexes Flash files, so this would theoretically not block indexing or scraping of content:

googlewebmastercentral.blogspot.com/2008/06/... – Dave R.
Dec 28, 2008 at 23:49

Flash will not only annoy you, but it will also get you removed from Google as well - Think again! – Fery Kaszoni May 16, 2015 at 14:27



0



I suspect there is no *good* way to do this.

I suppose you could run all your content through a mechanism to convert text to images rendered using a CAPTCHA-style font and layout, but that would break SEO and annoy your users.



Share Follow

answered Dec 29, 2008 at 21:50



Adam Jaskiewicz

11k ● 3 ● 36 ● 37



0



Well, before you push the content from the server to the client, remove all the `\r\n`, `\n`, `\t` and replace everything with nothing but a single space. Now you have 1 long line in your html page. Google does this. This will make it hard for others to read your html or JavaScript.

Then you can create empty tags and randomly insert them here and there. They will have no effect.

Then you can log all the IPs and how often they hit your site. If you see one that comes in on time everytime, you mark it as robot and block it.

Make sure you leave the search engines alone if you want them to come in.

Hope this helps

Share Follow

answered Jul 12, 2009 at 4:06



Avid Coder

18.4k ● 14 ● 64 ● 68



0



What about using the [iText library](#) to create PDFs out of your database information? As with Flash, it won't make scraping impossible, but might make it a little more difficult.

Nels

Share Follow

answered Jan 15, 2010 at 18:55



Nels Beckman

20.5k ● 3 ● 27 ● 28

1 no it will actually make it EASIER to parse, PDF files are really well supported for searching and indexing their text contents so all I need to do is download the .pdf files and process them off line at my leisure. – user177800 Jan 15, 2010 at 19:08



0



Old question, but- adding interactivity makes screen scraping much more difficult. If the data isn't in the original response- say, you made an AJAX request to populate a div after page load- most scrapers won't see it.

For example- I use the mechanize library to do my scraping. Mechanize doesn't execute Javascript- it isn't a modern browser- it just parses HTML, let's me follow links and extract text, etc. Whenever I run into a page that makes heavy use of Javascript, I choke- without a fully scripted browser (that supports the full gamut of Javascript) I'm stuck.

This is the same issue that makes automated testing of highly interactive web applications so difficult.

Share Follow

answered Jul 9, 2010 at 20:24



Matt Luongo

14.8k ● 6 ● 55 ● 64



0



I never thought that preventing print screen would be possible... well what do you know, checkout the new tech - sivation.com. With their video buffer technology there is no way to do a print screen, cool, really cool, though hard to use ... I think they license the tech also, check it out. (If I am wrong please post here how it can be hacked.)

Found it here: [How do I prevent print screen](#)

Share Follow

edited May 23, 2017 at 11:45



Community Bot

1 • 1

answered Aug 25, 2010 at 11:54



Tom

1

-
- 1 unrelated. Print Screen != Web Scraping. Print screen gets a snapshot of the screen , with apps and what not(like a BITMAP). Web scraping is the act of getting the data out of a web page into a form programs can work with.

– [Diego Castro](#) Dec 1, 2010 at 16:17

You let wondering on the work around (for the print screen). Wouldn't it be possible to emulate a machine, exec the prog on that machine and take a screenshot from the mother with the screen of the emulated on the middle (confusing???)

– [Diego Castro](#) Dec 1, 2010 at 16:46
