

SEVEN SIMPLE TECHNIQUES THAT VIRTUALLY ANYONE WITH ALMOST NO EXPERIENCE CAN USE TO HACK AROUND YOUR SHOPPING CART AND STEAL YOUR REPORTS



Andy Stetzing

Information and recommendations made in this report are for entertainment and informational purposes only. Any use of this information in violation of any federal, state and/or local laws is prohibited. Past performance does not guarantee future results. Any reliance on such information and recommendations is at the sole discretion and risk of the reader.

Table Of Contents

Introduction	3
Method 1: View Source	4
Method 2: robots.txt	6
Method 3: Variable Engineering	7
Method 4: Recursive Web Suck	9
Method 5: Search Engines	11
Method 6: Disposables	12
Method 7: Social Engineering	13
Conclusion	15

Introduction

The business of report writing is a huge one. There is always going to be a market for people wanting to share the information they know with the people who are willing to pay for that information.

Unfortunately, there are also scores of people who are not willing to pay for such information, and feel the need to circumvent and bypass the system in order to gain the information they seek.

“Information should be free” is their common mantra. They view themselves not as hackers, but as modern day Robin Hoods, sharing their findings with like minded people, who then in turn also download a copy of your report for free.

You worked hard on your report. It cost you time and money. The information contained within it is of value to you, and to your honest readers. If you require people to pay for your report or not is irrelevant – your report still cost you something, and you are certainly trying to gain something in return for your report; an email address, contact information, opinions – whatever it is, your report isn’t free.

In the next 7 chapters I’m going to show you techniques that I have personally used to gain access to reports and information. I’m also going to share with you how to keep others from doing the same.

For what it’s worth – when I do come across a security hole that allows me to download or receive a report for free, I do attempt to contact the owner of the report and let them know exactly what I did, and how to prevent it.

All the examples you will see are “Real World”, and currently online as of this writing. I have changed the names of the websites to “example.com” for obvious reasons.

Method 1: View Source

Common web browsers all contain the ability to show the html output of a page in a “raw format”. This will reveal what the page is made up of, including things that the author may not want me to see. This is the first thing I do when I find a “squeeze page” asking me to register or pay for a report.

What I look for in the source of the page is a FORM tag. This is what asks the viewer to input their information to get your report. More often than not, it will contain a “hidden” field called “redirect” which tells the browser where to go when the person is signed up.

```
<form method="post"
action="http://www.example.com/scripts/addlead.pl">
<input type="hidden" name="meta_web_form_id" value="XXX">
<input type="hidden" name="meta_split_id" value="">
<input type="hidden" name="unit" value="writingreport">

<input type="hidden" name="redirect"
value="http://example.com/thankyou.html ">

<input type="hidden" name="meta_adtracking" value="">
<input type="hidden" name="meta_message" value="1">
<input type="hidden" name="meta_required" value="from">
<input type="hidden" name="meta_forward_vars" value="0">
</form>
```

At this point, all I do is put the REDIRECT value into my address bar in my browser, and more than likely, I have access to the report.

The same holds true for a very popular way of allowing people to purchase your reports using PayPal. Simply view the source, look at the return value, and go to that location.

```
<form action="https://www.paypal.com/cgi-bin/webscr"
method="post">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="business"
value="paypal@example.com">
<input type="hidden" name="item_name" value="Special Report">
<input type="hidden" name="item_number" value="12345">
<input type="hidden" name="amount" value="89.95">

<input type="hidden" name="return"
value="http://www.example.com/paypal_thanks.htm">

<input type="hidden" name="cn" value="Comments?">
<input type="hidden" name="currency_code" value="USD">
```

In the above example, I received a report valued at \$89.95 for absolutely nothing!

Resolving both of these techniques is fairly simple. Most web servers these days support either PHP or ASP (both are server sided programming languages). They are both capable of identifying where the user arrived from, if anywhere at all. In both cases, the easiest thing to do would be to check if a the HTTP_REFERER value is set (in other words, did the user come from another web page?) If it's not set, it means they typed in the URL manually, and you should reject them. Furthering the security feature would simply be defining WHERE the user should have come from.

As for the PayPal button issue, simply select an encrypted button from PayPal's website – it doesn't reveal the return address information at all.

Method 2: robots.txt

When I cannot get enough information from viewing the source of a page, my next step is to view the robots.txt file (if one is present) on the server.

The robots.txt file is a simple text file that gives search engines a little bit of information about your website to make their job easier. It can tell different search engines how to index your site, how to handle images, where your sitemap is, when it should come back to visit next, and how fast it should roam around your site. It also tells them what NOT to index – and this is what I am interested in.

```
# Disallow directory /reports/  
User-agent: *  
Disallow: /reports/
```

At this point, I simply visit <http://www.example.com/reports/> and see what I find. Most of the time, I land on a page that says “Thank you for buying our report”, and can download the report simply by clicking on the link. In a lot of cases, a feature known as “Directory Listing” is open, and I can see the entire contents of that directory (or folder) – this opens up a whole other realm, which we will discuss in Method 4.

To avoid this, simply do not list the information in your robots.txt file. The only way it will get indexed is if someone links directly to it from their website, or if you link directly to it on yours. A more advanced technique would be to implement a username and password authentication routine for gaining access to that directory.

Method 3: Variable Engineering

For the most part, Methods 1 and 2 get around most of the free reports, and a pretty good deal of the paid reports. However, there are a number of sites that tend to employ a slightly more sophisticated method of access prevention by using a script or program.

In many web-based programming languages information is passed along by using what's called a "variable". This is a defined element that can be assigned a particular value. For example, you might see something in an website address that says:

<http://www.example.com/index.php?page=welcome>

In this case, the variable is "page" and it's value is "welcome". This would be telling the index.php file to, in essence, fill in the blank next to "page" with that information – "welcome".

A great deal of programmers will make variables "human readable" so that they can understand them when they're programming. So, taking the example a little further...

<http://www.example.com/index.php?page=reports>

Following the above URL takes me to the reports page.

<http://www.example.com/index.php?page=reports&report=1>

I am now looking at just one particular report.

<http://www.example.com/index.php?page=purchase&report=1>

This brings me to a page where I am supposed to pay for the report. Now, we simply follow the logic the programmer has already used, and continue the pattern.

Instead of filling out the information with my credit card and personal information, I type the following into the address bar:

<http://www.example.com/index.php?page=download&report=1>

I am now presented with the report, and have the ability to download it, without ever having given any information, or paying for anything.

Furthermore, by simply incrementing the report number (report=2 , report=3 ...) I am able to access the sites full range of reports.

Protecting yourself against this is still a simple process. Use variable names that aren't easy to guess, and use the HTTP_REFERER method we previously discussed.

Method 4: Recursive Web Suck

This is one of my favorite techniques. For years now, I have used a program called wget almost daily. It has great benefits in my programming life, but can also be used to steal valuable reports from your website.

In Method 2: robots.txt, I talked about directory listing. This is often left on as an oversight, and can be the source of great loss to the owner of a site selling reports.

Consider the following partial directory listing:

<u>Name</u>	<u>Last modified</u>
 Parent Directory	
 6 Viral Mini Videos/	27-Dec-2005 21:27
 13 Niche Products Pr..>	27-Dec-2005 21:27
 101 Business Ideas/	27-Dec-2005 21:27
 Add2it Leads/	27-Dec-2005 21:28
 Adsense Empire/	27-Dec-2005 21:35
 Affiliate Marketing ..>	27-Dec-2005 21:39
 Audio Player Pro/	27-Dec-2005 21:40
 AutoResponder Deluxe/	27-Dec-2005 21:40
 Best Articles Ever W..>	27-Dec-2005 21:40
 Blogs for Fun and Pr..>	27-Dec-2005 21:40
 Building better webs..>	27-Dec-2005 21:40
 Computer Tips/	27-Dec-2005 21:40
 Easy PDF Tool Kit/	27-Dec-2005 21:41

There are so many directories, each containing reports or programs, that it would take a person hours to download them all.

With wget, it is one command:

```
wget -r http://www.example.com/ebooks/
```

The `-r` tells wget to perform a “recursive web suck”. In other words, “go out and grab everything you can from this website.”

After I type that in, I sit back and watch a bunch of reports being downloaded to my computer.

It looks a little something like this:

```
--09:58:23--
http://www.example.com/ebooks//Million%20Dollars%20Ideas/362milliondollarideas.pdf
=>
`www.example.com/ebooks/Million%20Dollars%20Ideas/362milliondollarideas.pdf'
Reusing connection to example.com.com:80.
HTTP request sent, awaiting response... 200 OK
Length: 190,066 [application/pdf]

100%[=====>] 190,066
169.82K/s  ETA 00:00

09:58:24 (169.82 KB/s) -
`example.com/ebooks/Million%20Dollars%20Ideas/362milliondollarideas.pdf' saved
[190066/190066]
```

The only way the owner of the site would know anything had been downloaded would be to carefully check their web logs for access. Sounds easy enough, but when is the last time you checked your web logs?

Prevention of this is one of the easiest things a person can do. Simply put a blank index.html file in the directory, or have your system administrator turn off directory listing.

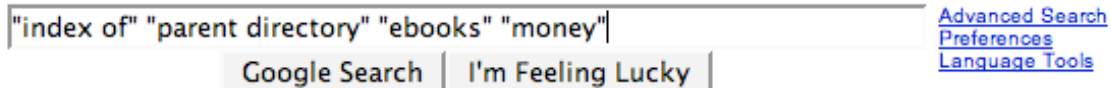
Incidentally, there were a total of 93 reports and software in that directory, totaling over \$8,910 in value. The complete download took 7 minutes and 34 seconds.

Method 5: Search Engines

This would be, by far, my most favorite information gathering tool of all time. It is often the gateway that leads to all the other methods described in this book.

The concept is simple. Search Engines live to give information away, and people aren't protective enough about what information ends up on the search engines.

Feel free to try the following "Basic Search" on Google.com (type it exactly as shown)



The terms "index of" and "parent directory" both appear on a page who's directory listing is enabled. The terms "ebooks" and "money" just narrow the search field. So, if Google's indexed it (which you'll see it has thousands of them) you'll find it there.

Once you find what you're looking for, Method 4 makes it quite simple to obtain.

If you happen to find YOUR site listed in Google, you can use the Google Webmaster Tools to remove your URL.

This method is not just used for finding ebooks – with the right search phrase on Google, you can find passwords, logins, log files (with good info in them), sensitive information, budgets, personal or private information... you name it! (Be careful what you search for, because you WILL find it)

Method 6: Disposables

There are a number of times where a webmaster has taken all reasonable precautions in protecting their reports and leave little on the table for me to work with. This is where the world of “Online Disposables” comes in.

For example, if I really want a particular free report, and none of the previous methods have worked, I simply use a disposable email address. I can grab a free disposable email address in a matter of a minute or so, and use it to get my free report. The disposable email address will expire in a matter of hours, but by then, I’ll already have everything I need.

Now I won’t be added to a mailing list, or have given the owner any of my real personal information.

An added bonus is that quite often, the email I receive does NOT contain the report I wanted. Instead, it contains a LINK to the report, which sometimes allows me to use some of my other Methods to obtain more information. People are well aware that most “Free” reports will lead you to purchase a product or a full report.

While this one is harder to prevent, there are lists of disposable email address services available. This is something that must be checked and updated at least weekly.

This is not limited to email addresses alone – there are disposable phone numbers as well, which I have used in my next Method.□

Method 7: Social Engineering

Social Engineering is the art of fooling someone into believing something that isn't true. It's lying on a professional level.

One of the common elements that most online marketers have is the desire to make as much money as possible, with the least effort possible, in the quickest time possible. Knowing this, a simple conversation with a webmaster can yield a great wealth of information and resources.

For example, I can easily setup a website that looks like it's doing well, and has a lot of visitors and traffic. I then find the contact information for the webmaster who has the information I want to obtain. I contact them via email (not a disposable one, but one I setup with my bogus website) and ask them if they would be interested in a trading some traffic and information

Once I have established an email dialog with the webmaster, I let them know about some of my "valuable reports" (the ones I obtained using other Methods listed here). I tell them I would like to offer their paid report as a bonus, and will likewise do the same for them, allowing them to have one of my paid reports as a bonus for their subscribers.

At this point, I simply ask for the report.

Sometimes they'll want to talk on the phone – this is where the disposable phone number comes in quite handy. We'll talk, get to know one another, whatever it takes to gain their confidence.

Then I ask for the report again.

If they want to work out a deal where I drive traffic to their site instead of allowing me to present my users with the report directly, I can come up with a dozen or so reasons why it isn't working for me. I can even allow their webmaster to look into the deal, and can guarantee you they won't find out why it's not working. This will lead them to allow me either to hotlink the report, or they will allow me to download it. This whole process should take less than 5 days to complete.

Preventing this method is a matter of doing your homework. Never accept a deal on face value. Examine the website, look at their history, Alexa stats, web logs, etc. Make them prove who they are, and that they are legit. Make sure you know some people in common, and find out that others say about them. If it's a serious relationship, ask to meet in person. Above all else, do not allow them to hotlink to your report. Insist on an arrangement that drives traffic to your site for registration purposes.

Conclusion

The seven methods I've shared with you are quite basic, and all of them are preventable with a little work on your part. There is not, and will not be a "catch all" way of preventing someone to steal your work. The irony is, I am sure this report will end up being stolen and passed around as well, despite my best efforts to prevent that from happening. It's a loss I am willing to accept.

There are other methods that I use to gain access to the information or report, but many of them require specialized programming techniques, and, quite honestly, I wouldn't want to give away all of my secrets in one report, now would I?

Thank you for taking the time to purchase and read this report, I do hope it was beneficial to you.

While this report shed light on the more popular ways of getting around squeeze pages, I can assure you that if you do nothing more than read this report, it will do you no good. You must take action on the methods described in this report for it to be of any value to you. I promise, it won't hurt you, and can only result in good.