

How To Think Like A Spammer...

So You Can Protect Your Online Forms

By Jack Born

Creator of "[Ultimate Form Mail](#)"

One of the things that you learn when setting up a new website is to make sure that you set up a mail reply option that minimizes spam. The Ultimate Form Mail program is great for accomplishing this. In addition, the script is customizable so that your website visitors can receive a customized reply from you or error comments if they do not complete the email as you have specified.

Richard J Nason
www.siteometrics.com

"We use Ultimate Form Mail for our clients, throughout North America, who require an automated form processing solution. Ultimate Form Mail is a rock solid solution and we get great, timely, support".

Garry Brownrigg, President
[SohoPortal.com](http://www.sohoportal.com)
<http://www.sohoportal.com>

After searching the internet and trying out dozens of form mail scripts, I was relieved to finally come across Ultimate Form Mail. This script does everything I need it to, is easy to use, and comes at a remarkable price. Thank you!

Troy Mumm
Third Sun Productions

Nice response time! Great product, by the way. I'm sure I'll be ordering more as projects come in. Thanks!

Jamie Hutto
Concentric Design & Consulting
(901)240.4951
www.concentricdesign.cc

Just wanted to let you know that Ultimate Form is the biz!
It's the only form that does everything I need - and securely.
Will be in touch soon as I'll be purchasing my fifth licence shortly (and I know you give discounts) :o)
Hope you are doing well.
H.
--
Huw Davies
huw@ebme.co.uk
<http://www.ebme.co.uk>

No Programming Code Discussed Here!

I wrote this guide for all skill levels... novice to advanced programmer.

Some of the concepts don't require much technical skill while others require some medium to advanced programming code.

“Ready to Use” Programming Code Available

I have copies of all the programming code you need to turn the concepts I am going to teach you into reality. All I ask is that you join my mailing list and let me give you a few ongoing pointers and updates about security and software to make your website run more smoothly.

You can get your copy of “ready to use” programming code you need to protect your online forms here:

[Free “Ready to Use” Code -- Click Here](#)

The Author: Jack Born

I make my living teaching entrepreneurs how to make the most of their online marketing. I'm one of the few web designers with a background in sales and marketing.

I started writing programming code shortly after I told my boss what I thought of him and found myself working harder than I ever imagined.

One of my earliest projects was creating a form processing program in PHP that was secure and simple to use. I reluctantly decided to take on the project when I realized there were no realistic alternatives for secure and simple to use form mail processors.

All I wanted was a form processor that could:

- ❖ Give me a great deal of security
- ❖ Was easy to use
- ❖ Was extremely flexible
- ❖ Could upload multiple attachments
- ❖ Save the information to a database
- ❖ And send a personalized reply to my visitor

The software I created is called Ultimate Form Mail and has been used successfully all over the world. From the very beginning, security was the highest priority.

Spam is a very big issue these days and it's getting worse. I hope my guide will give you some ideas about how to protect your website and your reputation so you don't have to come face to face with the "dark side" of the internet.

Whether you use my software, someone else's, or you write your own, you can use my suggestions to strengthen any online form on your website.

To your success,

Jack Born
[Ultimate Form Mail](#)

Your Online Forms... The Threat And What's At Risk

When a spammer sets out to send their unwanted emails he understands that someone's going to pay a price for his actions. People don't like spam and when they get spam they often report it to their ISP's (internet service providers) as well as other companies that maintain lists of spammers on the internet.

Whoever gets blamed for the spam (right or wrong) gets "blacklisted" by internet service providers like AOL, Earthlink, and so on. When you're blacklisted, you have trouble getting your emails delivered.

Since spammers want to keep sending spam, they think it would be best if someone else took the blame for their digital junk mail... you.

One way they get someone else to take the blame is that they find websites with online forms (feedback forms, contact forms, surveys, etc.) that are poorly protected. They then use some technical tricks to make an ordinary website send out a LOT of unwanted email... and guess whose return email address is on the spam?

AOL, Earthlink, and Hotmail aren't going to invest resources to determine exactly who is to blame for the spam. If a flood of spam is coming from your website, they're going to blacklist you, and they may very well blacklist any website associated with your server. They have to protect their members from spam or they'll lose business to someone that will protect them.

When this happens, **don't be surprised if your web hosting company decides to shut your website down without notice.** Your web hosting company has a business to run and they need to protect their other customers. If something you put on your website is a threat to their business... you're history.

Think Like A Spammer To Protect Your Web Forms

It's no wonder that web designers and programmers usually have a difficult time thinking like a spammer, or anyone else with plans to twist their creation for unintended purposes. We create websites and software for productive reasons. We want to sell, to teach, to provoke, and to reach out. Accomplishing our goals on the web is difficult enough without having to think about how our work could be used against us.

But thinking like a spammer is exactly what you need to do if you're going to protect yourself from having your website used as a vehicle to send a mountain of spam. If you are caught with your guard down, your web hosting company

may shut down your website, your email addresses may get blacklisted, and it's possible that all the other websites sharing the server with you will get blacklisted as well.

As the saying goes, an ounce of prevention is better than a pound of cure.

Forget The Technical Details

A lot of advice on the web talks in great detail about how specific spam attacks are carried out. I think getting bogged down in the details can confuse the issue and make it seem too difficult to defend against spam when it's really not that tough.

If you stick to the basics and just put ourselves in the shoes of a spammer you'll take a huge step towards securing your site. This approach is easier to understand and makes your decisions for defense very straightforward.

First, let's think like a spammer for just a minute...

Four Steps To Successful Spamming

1) To be a good spammer you need to find some online forms... a whole bunch of them

If your online form is easy to find then your chances of getting attacked are much higher.

2) Next, quickly sort through the forms you find and find the ones that look like the easiest to go after...

A guy I know that fishes in Alaska told me that he once asked a fishing guide why he didn't seem very concerned about the chances of spotting a grizzly bear. The guide's answer was, "Because I know I can outrun you."

There's probably no such thing as 100% security. But if you give a spammer a choice between working hard to crack your online form or going after hundreds of easy targets somewhere else... you're improving your chances of staying out of trouble.

Don't be an easy target.

3) When you find forms that are vulnerable, send your message to the unsuspecting website form processor with a BIG list of emails that will receive the spam

The very definition of spam implies **massive** amounts of unwanted **emails** sent out to a large number of email addresses belonging to people that never asked to receive solicitations from the sender.

You're going to make it really tough on a spammer if you:

- ❖ Limit the amount of data you'll accept
- ❖ Limit the occasions where you'll accept emails to exist in the data
- ❖ Limit the number of emails allowed on the rare occasions you allow email data

4) Automate the process. It's no fun doing this work by hand, for heaven's sake.

Most spammers find, test, and attack websites *remotely* using automated programs. In other words, they're not visiting your website (in most cases). They can launch an attack from their desktop or from their own websites.

If your online form makes it very, very difficult for someone to test and attack remotely then you've eliminated an enormous percentage of your potential attackers.

It's very unlikely that any one spammer will single you out and spend a great deal of time and energy visiting your site, testing for possible weaknesses, and persisting through multiple failed attempts. They don't have a grudge against you and this is nothing personal. They're looking for easy targets.

Now Let's Play Defense

Knowing what spammers are looking for and the basic concept behind spamming, we can craft some very simple but effective defenses against most spam attacks.

For you hard core programmers out there, you may see that some of my methods are a little more simplistic than you'd like to see. I'm well aware that there are some more elegant programming solutions to spam but I'm outlining a common sense approach that anyone can use... not just us "propeller heads".

Defense #1: Hide your online form

No matter whose form processing code you use, you shouldn't name the web page with your online form something easy to find.

Bad names for your online form web page include anything with the following words in them:

contact
form
feedback
mail

Which includes...

contact_us
feedback_form
mail_form
form-mail

And so on.

The idea is simple: if they can't find it, they can't attack it.

Be careful! If renaming files is your only defense, then it's only a matter of time until you're found and attacked.

Using this as your only line of defense is a horrible idea. But... used as one step in your overall defensive strategy, this is an excellent idea.

So, we need to add some more defensive measures in case they find your form.

Defense #2: Remove instructions from your html

The easiest exploit that a spammer can spot from a mile away is seeing a hidden field in your form that lists the intended recipients for the form data.

If the techno-speak just threw you for a loop, don't worry... here's the English version:

Huge Tip:
Look at the source code for your form and if you see your email address in the source code... you've got a big security problem waiting to happen.

Looking at the source code is easy. Just right click and choose "View Source". A bunch of mumbo jumbo (HTML) will come up in a text file. At the top of the page you'll see 'Edit' as one of the options. Click 'Edit' and then choose 'Find'. Or, on some computers you can just hold down Ctrl and 'F' for Find. Type the '@' symbol in the Find field and search. Keep searching forward in the document until everything with an @ symbol is found for you.

If you see an email address and very close by you see...

type="hidden"

...you probably have a form that is not secure.

A good, secure form processor has a predetermined recipient. In other words, it knows where to send the form data before it even starts processing the data. The target email(s) is already hardwired into the code of the form processor.

On the other hand, a form that tells the programming code what email should receive the information only *after* the form Submit button is clicked is very dangerous and easy for a spammer to abuse.

If your email address is in a "hidden" field in your html code then your form processor is waiting for the instructions to tell it where to send the email. That's not good. You want the target email address (recipient) written right into the code of your form processor. (By the way, a "hidden" form field is very easy for spammers to find. It might as well be in bold at the top of your page.)

Defense #3: Say "NO" to big lists of email addresses

Here's where the real defensive measures come into play. Unfortunately, this is also the step where some computer programming comes in...

Wait!... I'm not going to show you a bunch of programming code. I promise.

You don't have to know any code, learn any code, or hire anyone to code for you. For one thing, I'm not going to get into the technical details of how to write the code to make these concepts work. And if you're not the type to write your own programming code (or you're just looking for a shortcut) then I've put together some [code you can use on your site right away](#).

We're going to keep it real simple. Even if you're an advanced programmer, you might discover a new trick or two by looking at the problem from a new angle.

To figure out how we're going to prevent spammers from sending a huge list of email addresses over to your website **we're going to focus on three details that are so obvious that many people overlook them entirely.**

Obvious (but often overlooked) fact #1: A spammer wants to send out a message to a LOT of people

Obvious (but often overlooked) solution #1: Limit the amount of characters in a data field whenever you can

Every data field should be specifically limited in terms of the number of characters you are willing to accept. Anything over that number should be discarded.

Most of the information you're asking for in your form is a short answer: state, city, zip, name, birthday, favorite color, etc. So limit the amount of data you'll accept and instantly you make it really difficult to send over even two email addresses, let alone several thousand.

Obvious (but often overlooked) fact #2: All email addresses have the @ symbol

Obvious (but often overlooked) solution #2: Sanitize data that shouldn't contain email addresses by removing the @ symbol

If you have ten data fields in your form it's very common that nine of them don't ask for email information. So, you should have a process in your code that automatically purges the @symbol from each data field that comes over. This data cleaning process should be the rule, not the exception. What I'm suggesting is that you should cleanse all data except the one or two fields that you know should be email addresses.

Obvious (but often overlooked) fact #3: It's very rare that an online form asks you to fill in MULTIPLE email addresses. When an email address is requested, it's almost always ONE.

Obvious (but often overlooked) solution #3: When you ask for one email address, make sure there's only one sent over.

I've never seen an online form that asked for me to type in multiple email addresses into one form field. (Don't bother telling me that they exist... I'm sure they do, they're just not common.)

Since 99% of all forms that ask for an email address only want an answer with one email address, it makes sense to verify that only one email is sent through... not two, not three, and certainly not 500.

It's safe to ask for an email address if you have a good email verification function to throw out bad emails (such as long lists of emails) or... you could count the number of @ symbols and stop the code if you find more than one.

Warning! JavaScript Will NOT Protect You

JavaScript is a great tool to make it easier for the good guys to send you the information you need and to help them fill out your form with the least amount of trouble.

JavaScript is NOT a tool for protecting your form from spammers. You need to have protective measures that work on your server. Whether your code is written in ASP, PHP, Perl, Python, or Cold Fusion you need to make sure that your defense mechanisms run on the server, not your visitor's browser.

JavaScript can be turned off very easily. But more importantly, most spammers find, test, and attack your website without ever visiting your website. Which is a perfect lead in for the last defensive measure:

Defense #4: Force them to do the work by hand

As I just mentioned, most spammers want to find, test, and attack your website from a distance. It's not that they're afraid of visiting your site... it's just a whole lot easier to automate the spam process than do it by hand.

Even if a spammer doesn't share your ethics and values they do share your instincts. We all operate somewhat predictably according to "human nature". One of the first rules is that we all like to take shortcuts. We're inherently lazy. "Why work harder than we have to?" is a question echoing in all of our brains.

So, if you force a would-be-attacker to visit your website personally to test and probe for weakness then you're making it much more difficult for them.

"How do I force someone to fill out my form online?"

Glad you asked.

There are probably several ways to almost require anyone sending data through your form to do it from a browser, the way you always intended. I'll show you two ways. The first one is very common, but I prefer the second one I'll show you.

Captcha - Completely Automated Public Turing Test to Tell Computers and Humans Apart

You've undoubtedly used an online form where you were required to interpret a few squiggly letters with shapes in the background or foreground. This test is designed to tell computers and humans apart and it's called a CAPTCHA.

Some very popular online services use them including Yahoo, Paypal, Earthlink and Hotmail. But I don't recommend them.

For one thing, they're annoying for your visitors. A certain percentage of your visitors who otherwise would have sent you the information you wanted will choose to skip the effort if you require them to fill in a CAPTCHA. People are lazy... remember?

In addition, software has been created to circumvent the CAPTCHA by reading them. Some software test have reported over 80% success at reading various CAPTCHAs.

Token-Session Matching

I first read this concept from an article written by a PHP security expert named Chris Shiflett. You generate a unique "token" or random alphanumeric string, store the information in a "session" (a specific type of cookie) on your visitor's computer and then compare the two variables when data from the online form is sent to be processed. If the information in your visitor's browser (the session) matches the token data sent over, then all is well. If not, then the program is halted.

The reason I like this approach so much is that it makes thing easy for the good guys (they don't have to do anything different at all) and it makes things tough for the bad guys (they have to visit your site to even attempt to exploit it, and they're limited to a short time frame before they have to visit your site again).

Resources

Recommended Software

[Ultimate Form Mail](#) - Free trial version.

Articles

[Chris Shiflett](#) - great articles on security issues

[Secure PHP](#) - a technical article on email injection threats (just one way a spammer can attack you)

Code

[“Ready to use” code](#) to turn the concepts in this report into reality.