PHPit Totally PHP

Clickbank Security Using PHP

By Dennis Pallett

Here's a way to protect the products you sell with Clickbank, using their built-in protection and by implementing a 30-day expiration, all without having to worry about managing databases or customer lists.

The first step

First of all, Clickbank protection is decent as it is. If you want to keep your customers from passing the thank you page URL around to friends, there are a couple of things you can do.

Login to your Clickbank account: http://www.clickbank.com/login.html

At the top there's a link that says "Click HERE to modify your account". Click on the link.

On this page there are two links at the top, but one says "Click HERE to modify your account." Click on this one.

You should be at the page that allows you to edit the prices of all your Clickbank products. Scroll down to the bottom where it says:

Secret key (up to 16 letters & digits)

You should see a text box here. If it's empty, choose a secret key, type it in and remember it. It can be anything you want, but it should be different than your Clickbank password.

If you've looked around the Clickbank site you'd know that Clickbank offers some friendly pieces of code in a few different programming languages like Perl and PHP that can help you protect your downloads. Basically this is what happens:

* Your order link contains what's called a "seed". This is just a word or a piece of text, which can be anything you want.

* Your customer clicks on the order link and pays.

* Clickbank takes that seed, and uses your secret key on it -- basically mashes the two together and does a bunch of crazy stuff to come up with a garbled piece of junk. But this a garbled piece of junk that can ONLY come from this seed and secret key. You change the value of the seed or secret key even a little and this "hash" changes.

* The seed and the hash are passed back to the thank you page where your Clickbank script sits. (We have the secret key added to your script, and it never changes, so it doesn't need to be handed to us by

Copyright (c) 2007 PHPit.net

Clickbank.) This Clickbank script takes the seed and the secret key and does the same crazy shit Clickbank did to us to compute your own hash.

Clickbank calls this their "cbpop" or Clickbank Proof of Purchase.

The hash was something we figured out on your own and the hash Clickbank are compared. If they match, we're in business because the customer here really did buy from us.. The customer can't figure this out on his or her own because they never actually saw the secret key. (And no, you can't "reverse" a hash to figure out the original secret key.)

If you get nothing out of what I just told you, remember this: it's almost impossible for anyone to figure out the right Proof of Purchase code without that secret key.

This is the PHP function they give us:

```
function cbValid($seed, $cbpop, $secret_key) {
 // A bunch of stuff in here...
 }
```

This function cbValid takes three parameters: $seed, $cbpop, and $secret_key. The script goes through that last step of ours I explained above, does the crazy shit and then compares the result to the one given to us by Clickbank.

Now we need to figure out what to do if your customer really didn't pay. The easiest thing to do, is just stop the script in its tracks, preventing the page under it from loading.

```
if (!cbValid($seed, $cbpop, $secret_key)) die();
```

The exclamation point means "not". We're saying, first try this...

```
cbValid($seed, $cbpop, $secret_key)
```

.. pass the seed, proof of purchase, and secret key into your black box. If the function tells us NO, do the rest. In this case, "die". Die stops everything immediately, so if you have HTML or PHP code below that line, it won't be looked at if the Clickbank validation fails.

The "proper" way to grab $seed from the query string is this way:

```
if (!cbValid($_GET["seed"], $_GET["cbpop"], $secret_key)) die();
```

You could also redirect the user to an error page of yours if you like:

```
if (!cbValid($_GET["seed"], $_GET["cbpop"], $secret_key)) {
 header("Location:http://www.your.host/error.html");
 die();
 }
```

Instead of $seed and $cbpop we use $_GET["seed"] and $_GET["cbpop"]. This is because the variables don't appear magically out of thin air, they really appear in the URL as http://www.your.url/test.php?seed=SOMESEED&cbpop=SOMEPOP. We want these values to be taken out of the URL.

Here's a zip file containing your cb.php script: http://www.jumpx.com/tutorials/clickbank/cb.zip

Copyright (c) 2007 PHPit.net

Save it, unzip it, and open cb.php. Near the top should be a line such as:

$secret_key = "YOUR_SECRET_KEY";

Change YOUR_SECRET_KEY to that secret key you set in the Clickbank control panel.

Now, for usage... your thank you pages will have to end in .php here. Like, thankyou.php (and now it doesn't matter if they have obvious names or not -- because they'll be thoroughly inaccessible to thieves. Remember, you can simply rename your HTML pages so they end in .php and they'll still work just fine.

Put this line at the top of you thank you page script:

Be sure to upload cb.php to the same folder as your thank you page. Now, when someone goes to the thank you page, the first thing the thank you script will do is run everything in cb.php, and cb.php will take the data Clickbank has passed to see if it matches.

You're going to have to change your Clickbank order links a little. This is what they should look like now:

http://www.clickbank.net/sell.cgi?link=YOUR_CLICKBANK_ID/YOUR_PRODUCT_ID/YOUR_PRODUCT_NAME&sec

Replace YOUR_CLICKBANK_ID with, of course, your Clickbank ID and YOUR_SEED with the seed you want to use. This can be anything, something simple that's short and one word like the product name. But NOT your secret key.

YOUR_PRODUCT_ID is the number Clickbank shows to the left of each thank you page as you add it. When you're testing, be sure to set the price at $0.00. Once everything's in place you can raise the price of the item to $19.95 or $29.95 or whatever it's priced at.

http://www.clickbankguide.com/merchant.htm#account will explain everything if you're a Clickbank newbie.

You can't prevent sharing completely... after all, your customer can always download the file and share the file, not the download URL, to friends. We can do one thing to give these would-be freeloaders a bit of a headache, and that is expiration.

Here we can say, 30 days after someone buys your product, the thank you page will be inaccessible to them. If they buy on October 25th, they can bookmark and revisit that thank you page up until November 25th at the

Copyright (c) 2007 PHPit.net

exact time they made their purchase. It's kind of a nice compromise because it gives honest people enough time to get what they need but at the same time it becomes impractical to share the URL.

In chapter 9 of my book, Simple PHP (http://www.simplephp.com), I explained how time works on computers, they use a big number which is just a count of how many seconds have passed since January 1st, 1970. I also explained that there was a function, called strtotime(), which we could use to determine this "number" or timestamp of a certain date. For example, 30 days ago or 1 year ago.

30 days sounds about right. To figure out the Unix timestamp of this moment, minus 30 days is:

strtotime("-30 days")

Now, to store it in a variable called $expire:

$expire = strtotime("-30 days");

But you're saying, how do I know when these people purchased? I don't have that kind of information. Aha! But you can. Remember, the seed you put in your order links can be anything you want. So let's just make it the timestamp of this exact moment.

When the customer revisits the thank you page, they can't change the seed, because as I mentioned, if you change *either* the seed or the secret key, the resulting hash (proof of purchase) will be different. So you see, they're stuck with it. But, the current time always changes!

All we have to do, in cb.php, are these two steps:

* Figure out what the timestamp was exactly 30 days ago, and store this value in $expire.

* Compare the seed and $expire. If the the value of the seed is less than that of $expire, it means that the product was purchased more than 30 days ago and the visitor shouldn't be given access to the page. Die.

We've already taken care of step one by saving the timestamp 30 days prior in $expire. Now, we compare the seed (it's $_GET["seed"], remember, because we're grabbing it out of the URL string) and $expire like:

if ($_GET["seed"] Order Now

Instead of YOUR_SEED we want PHP to call the function mktime(), which gives us the current timestamp, and output it, using echo.

echo mktime();

Now just put around it...

And shove it in there.

">Order Now

Now setup a link for $0.00 in your Clickbank control panel and try it. You can be sure it works by changing that "-30 days" in strtotime to "-5 minutes". Then try accessing the download page, then wait 5 minutes and try again. Neat, isn't it?

You say, I've done this, but I have more than one product. How do I keep someone from grabbing everything once they've grabbed one?

Copyright (c) 2007 PHPit.net

Have your links look like the following: ">Order Now

This way the seeds will look like "stringbeans445433" if you're selling stringbeans. Then again if you're selling corn on the cob on another sales page, you can change "stringbeans" to "cornonthecob". Now the seeds for each product will be different.

Those seeds won't be all numbers, will they? So, in cb.php, do this:

```
$timestamp = ereg_replace("[^0-9]","",$_GET["seed");
```

I won't go into a lot of detail about pattern matching, but the [^0-9] means "NOT anything from 0 to 9. It basically goes through every letter and number of $_GET["seed"], and if what's there isn't a 0, 1, 2, etc. it's replaced with nothing (hence the ""). The final result is saved in a variable called $timestamp.

Since now we're looking at $timestamp and not $_GET["seed"], let's change that if-statement:

```
if ($timestamp
```

When I extracted the timestamp from the seed, I simply removed all characters that were not numbers, leaving just the numbers contained within that string. Now I want to do the opposite. Here's an example seed:

test1074482258

I take out all the numbers and am left with "test". Next I figure out which script called cb.php (which is stored in the variable $_SERVER["SCRIPT_NAME"]). Then the script takes out everything up to the last slash (/) and everything before the first dot (.). If the script was located at "/clickbank/test.php", all that's left is "test".

If you give each thank you page a different name, and make sure all your seeds reflect the correct page, i.e. if your thank you page is called "carrots", the part of that order link containing the seed should appear as:

```
&seed=carrots
```

If you don't care how Clickbank's protection works, that's your derogative. Just get the zip file and follow the instructions I've provided in cb.php.

As far as scripts that handle several Clickbank products -- I can't recommend any at this time, since I've never across any good ones. (But you should check out Harvey Segal's free site, ClickbankGuide.com, which can answer most of your questions about Clickbank.)

Here's that script again in case you missed it: http://www.jumpx.com/tutorials/clickbank/cb.zip

Make sure to read the instructions I've supplied in cb.php, get everything setup and on your web server, and you'll be well on your way to having bulletproof protection on your Clickbank products.

About this PDF

You may distribute this PDF in any way you like, as long as you don't modify it in any way. You can ONLY distribute the unchanged original PDF.

For more information, contact us at support@pallettgroup.com.

Copyright (c) 2007 PHPit.net