Could a Malicious Hacker Alter a Hidden Post Variable

Asked 13 years ago Modified 5 months ago Viewed 5k times Part of PHP Collective

37

I know that a POST can be spoofed in terms of originating domain, but what about being able to change the variables of the hidden POST variables in my HTML? I am concerned that someone could alter the "amount" value in my PayPal form from this:

```
<input type="hidden" name="amount" value="1.00">
```

to this:

```
<input type="hidden" name="amount" value="0.01">
```

or something similar. Thanks.

PHP php html validation post

Share
Improve this question
Follow

edited Nov 29, 2011 at 15:32

Andrew Kozak
1,660 ● 2 ● 22 ● 35

asked Nov 29, 2011 at 15:29

Vlad
475 ● 5 ● 9

11 This is easily possible with javascript. Even easier with the firefox web developer toolbar. You can make hidden form fields visible. – Gazler Nov 29, 2011 at 15:30

11 Consider also that there is NO REASON TO PRESUME that your form is being used to POST data to your server in the first place. – horatio Nov 29, 2011 at 19:20

2 `curl -d amount=0.01 http://path/to/form/handler` – SingleNegationElimination Nov 29, 2011 at 23:49

2 Heck, I know a *benevolent* hacker who could do this. – Jean-François Corbett Dec 2, 2011 at 8:25

6 Answers

Sorted by: Highest score (default)

Yes, it is trivially easy for anyone to modify your form variables. Whether they are GET or POST doesn't matter at all.

76

Web security rule #1: Never trust any user input. Also stated as "All users are malicious hackers" or some variant thereof.

answer to comment: The solution is to know all of the correct values on the server side, without having to pass them through the client side (Javascript). So regardless of what the form says, you already know the price. Just use the same value you used to populate the form in the first place.

Share

Improve this answer

Follow

edited Nov 29, 2011 at 15:36

answered Nov 29, 2011 at 15:31

Tesserex

17.3k ● 6 ● 71 ● 108

32 Emphasis on the word *trivially* – Bob Kaufman Nov 29, 2011 at 15:33

1 then what are my alternatives? only use php to create and submit the form? also, isn't this a security risk for all paypal merchants who don't use an encrypted button? (the reason I don't use an encrypted button is I need to change these variables on the fly) – Vlad Nov 29, 2011 at 15:34

7 @Vlad Why do you need the user to pass back something that he/she is not allowed to use? Can you not just store that info server side, or if you have to send it to the user, sign the data so you can check if it has been tampered with? – Roger Lindsjö Nov 29, 2011 at 15:38

11 Ideally, you wouldn't be taking amount as any kind of input. You would take a list of items, and you would calculate the amount in your php script. At the store, you would very rarely see clients fill their shopping carts, and then tell the cashier how much they have to pay. – Frank Nov 29, 2011 at 15:42

6 +1 for that web security rule. Never (completely) trust any input that came from the client, ever. This potentially includes *encrypted cookies*. All input must be completely checked and validated, *every* time it is recieved. Never store anything other than some sort of id/session token client side (which must still be validated). Note that some 'malicious' users may not be doing it deliberately, if their computer is part of a botnet. – Clockwork-Muse Nov 29, 2011 at 17:05

13

Update 2020:

OWASP covers this topic in "Injection Theory", where applications accept data from untrusted, uncontrolled, or potentially compromised sources.

> Injection is an attacker's attempt to send data to an application in a way that will change the meaning of commands being sent to an interpreter.

Review this OWASP "cheatsheet" for an overview of mitigations that can be implemented to better secure REST based endpoints.

Yes, it is very simple to do with browser inspector tools, JavaScript, cURL and other tools.

You shouldn't rely on the `amount` field being what you'd initially transmitted in the response to the client. A more secure approach would be to rely on an identifier for an item, which you can map to a price on the server (*a more controlled environment*).

Share
Improve this answer
Follow

edited Sep 30, 2020 at 22:13

answered Nov 29, 2011 at 15:30

Alex
35.4k ● 5 ● 79 ● 92

7

Yes, it is possible to change that value using javascript. If you haven't practice in using javascript, you can also do the test using Google Chrome's Developer Tools.

Infact this is one of the main reason to **don't rely on user input**.

Share Improve this answer Follow

answered Nov 29, 2011 at 15:32

Aurelio De Rosa
22.1k ● 8 ● 49 ● 71

7

Forget javascript and browser tools. Please realize that I can send ANY cookie, POST and GET argument (key and value pairs) I want, regardless of whether this is a form for them. (See cURL)

Frank said "At the store, you would very rarely see clients fill their shopping carts, and then tell the cashier how much they have to pay."

Try to think of it like that. The **browser** (not user) is the client and the server is the cashier. Any information that flows from the browser to the server can be anything I want.

Share
Improve this answer
Follow

edited Nov 29, 2011 at 19:45

answered Nov 29, 2011 at 19:23

user606723
5,105 ● 2 ● 30 ● 36

5

Yes. It gets worse because they don't even have to alter *your* page to do it. A user could use any text editor to construct an html page with a form full of text boxes, load it from local disk, fill them with whatever they want and hit submit. OTOH, that will show up in some header values.

Or if they are really determined, that can connect to port 80 on your server via telnet and forge the entire HTTP request including headers.

There is not a single byte of the incoming request that you can trust.

That said, there are known solutions to these problems that are generally implemented in terms of hashes, signatures and cryptography, but I don't know enough to suggest where to look for them.

Share Improve this answer Follow

answered Nov 29, 2011 at 18:26

BCS

78.3k ● 69 ● 194 ● 298

-4

In this case *let them* change the value if they want to, and let them pay you the $0.01. But then, when you get the Paypal IPN (Instant Payment Notification) it will include the price they paid; check that against your item database to make sure that it's the correct price.

If it's not the correct price, do not send/give the item. You earned $0.01!

Share Improve this answer Follow

answered Nov 29, 2011 at 18:35

Andreas Bonini

44.7k ● 31 ● 124 ● 158

haha. Would that be legal? Taking the law into your own hands? – Ian Warburton Nov 29, 2011 at 20:55

2 Dude; that would be theft. But if the OP is selling a download he should just send the % of the bytes for which the user has paid for :) – iHaveacomputer Nov 29, 2011 at 23:35

6 Best case: You earn $0.01 and pay out a few $k in lawyers fees. – BCS Nov 30, 2011 at 2:15

3 This post seems to have some merit en.wikipedia.org/wiki/Instant_payment_notification. There is a verification step in there, if the numbers don't add up then you would need to reject the transaction or something ... not keep the money. – row1 Nov 30, 2011 at 10:35

1 Strangely, this *is* the what PayPal expects you to do. At least for the moment, PayPal doesn't let you reprogram its servers, and they do look for the amount value coming from the customer's POST action. – Edward Newell May 19, 2014 at 21:36