The best way to secure posting data to a URL from Java

Asked 13 years, 11 months ago Modified 13 years, 11 months ago
Viewed 242 times

4

I am developing a small game (in Java) for a coursework and the extension I have decided to do is an online score board. As I thought might happen, a few people on my course have figured out how to hack the score board and submit their own scores.

I know there are a few problems in the current way of submitting scores, but here it is. The game generates a score, then does an HTTP GET on a URL with the options of the players ID, the score and a password.

I might changing this to be a POST as it might be more difficult to get the data for the password. Also, I am considering making it run over HTTPS (although I don't know if this is more difficult in Java). Unfortunately, this doesn't stop the main way that people found the password which was by decompiling the Java code.

I don't know the best way to prevent the hacking. I don't mind too much really, its not that important, but it would be nice to secure it so when the code is marked it doesn't have a load of spam on it.

What would be your suggestions on ways to obfuscate the code and/or secure the whole process?

java https obfuscation

Share

Improve this question

Follow

asked Jan 14, 2011 at 10:57

danpalmer

2,173 ● 4 ● 25 ● 42

What is your problem with POST? I think, POST will suit more! – Abimaran Kugathasan Jan 14, 2011 at 11:05

3 Answers

Sorted by: Highest score (default)

7

Your approach simply doesn't work. When you need to secure something, you can't run it on the client.

Instead, the whole game must run on the server and users can only submit moves. That way, you can validate the moves (so players can't create illegal states) and calculate the correct score.

Everything else can always be hacked. If you don't encrypt the data, it can be hacked by using a network sniffer. If you use HTTPS, hackers can use a proxy to decode the data (man in the middle attack).

Share Improve this answer

answered Jan 14, 2011 at 11:05

Follow

Aaron Digulla
328k ● 110 ● 622 ● 834

2 There's no need for a man-in-the-middle attack - users own the client already. They can just hack the client to do their bidding... – sleske Jan 14, 2011 at 11:21

1 What I meant is that you don't need to hack or decompile the client. Just install a proxy, install an SSL certificate and configure it for the client. That way, the client will send the encrypted data to the proxy which will decode and re-encode it. Script-kiddy level. – Aaron Digulla Jan 14, 2011 at 13:29

I know the server needs to know everything for it to be the most secure. However, the point of this question was more: what is the most effective with the least amount of work. I know that makes me sound lazy, but its more because of coursework deadlines. I was wondering if there was anything simple I could do to block the majority of simple attacks. Thanks for the answer anyway, the Man in the Middle Attack stuff was interesting. – danpalmer Jan 15, 2011 at 15:47

@danpalmer: Everything that you can do on the client can be broken by a skilled hacker in a few minutes. But here is a suggestion: Leave it as it is and award extra points for skilled attacks. I mean: You want to teach them how to use a computer. Breaking others code is a valuable skill, especially if you want to learn how to *protect* your own code.
– Aaron Digulla Feb 1, 2011 at 4:39

2

What one can do is changing from anonymous to user based tracking. This does not prevent faking, but makes it more trackable.

The basic protocol could be that a score board change is signed or encrypted using a session key. The session key is created upon logon of the player itself. Here you can work using an appropriate authentication system.

Now at least you know from which account a change has been made and can blame your student...

Share Improve this answer

Follow

answered Jan 14, 2011 at 11:19

mtraut

4,740 ● 3 ● 26 ● 34

+1 Good idea. If you decide to trust the client (rather than do everything server-side), you at least want to know who you are trusting... – sleske Jan 14, 2011 at 11:22

This is a good idea, I do like this a lot and may implement it. Thanks for the idea. – danpalmer Jan 15, 2011 at 15:50

2

Obfuscation will not help you. It will just make it harder for the hackers, but certainly still possible.

In order to secure the whole process (at least to some degree), you should not have any game logic code on the client. The client should only send game commands to the server. After receiving the commands the server

should check if the user can preform the given commands.

This way for example even if your colleagues send a command 'set jeff score 9999', the server would check if jeff has actually played a game that has given him 9999 points, if no, you could just show an error message on the client.

The general rule is: if something is on the client, the client can modify it. Most people conceder attempts to make it hard for the client to modify it futile, since it is only a matter of time before they do.

Share Improve this answer Follow

edited Jan 14, 2011 at 13:25

answered Jan 14, 2011 at 11:20

Simeon

7,782 ● 16 ● 67 ● 102

> Unfortunately, the entire coursework was about making a game that was local only. The extension (only worth 20%) was about adding some basic extra functionality. Putting the logic on the server would have been a lot of work for relatively few marks, and the server was not implemented in Java which was something we were being tested on. I am however taking note of all this advice for future projects.
> – danpalmer Jan 15, 2011 at 15:49