Open source web app more prone to hacking?

Asked 14 years, 4 months ago Modified 14 years, 4 months ago

Viewed 675 times

7

At a recent interview, I was asked:

Open source web app (say built on Struts/Spring) is more prone to hacking since anyone can access the source code and change it. How do you prevent it?

My response was:

The java source code is not directly accessible. It is compiled into class files, which are then bundled in a war file and deployed within a secure container like Weblogic app server. The app server sits behind a corporate firewall and is not directly accessible.

At that time - I did not mention anything about XSS and SQL injection which can affect a COTS-based web app similar to an open source one.

My questions:

a) Is my response to the question correct?

b) What additional points can I add to the answer?

thanks in advance.

EDIT:

While I digest your replies - let me also point out the question was also meant towards frameworks such as Liferay and Apache OFBiz.

java security web-applications

Share

Improve this question

Follow

edited Aug 23, 2010 at 6:30

asked Aug 23, 2010 at 6:12

newtoallthis

173 • 6

4 Answers

Sorted by: Highest score (default)

16

The question is a veiled argument towards Security through obscurity. I suggest you read up the usual arguments for and against and see how that fits:

- Security through obscurity (Wikipedia)
- Hardening Wordpress
- SSH server security (Putty)

My personal opinion is that obscurity is at best the weakest layer of defence against atack. It *might* help filter

out automated attacks by uninformed attackers, but it does not help much against a determined assault.

Share Improve this answer Follow

edited Aug 23, 2010 at 6:23

answered Aug 23, 2010 at 6:17

Robert Munteanu
68.2k ● 39 ● 209 ● 282

1 I agree with this answer. I'd also like to remind you that you will probably be using a vanilla installation of Struts, Spring and probably all of your Open Apps, so It's irrelevant that you are deploying Class files, as those came from the exact same original source code. – Sebastian Oliva Aug 23, 2010 at 6:25

Thanks Robert - I can see I totally misunderstood this aspect or indeed the question. – newtoallthis Aug 23, 2010 at 8:11

4 @newtoallthis - While this is an excellent answer, the question is likely motivated by a personal agenda. The person asking the question doesn't like Open Source and is trying to come up with reasons why. The argument they're presenting (through a leading question) isn't very supportable, but the basis for it is not rational so that doesn't really matter to the questioner. – Omnifarious Aug 25, 2010 at 15:30

5

> a) Is my response to the question correct?

The part about the source not being accessible (to change it) because it is compiled and deployed where it

cannot be touched is not a good answer. The same applies to non-open-source software. The point that was being made against an open source stack is that the source is accessible to read, which would make it easier to find vulnerabilities that can be exploited against the installed app (compiled or not).

The point about the firewall is good (even though it does not concern the open- or closedness of the software, either).

> b) What additional points can I add to the answer?

The main counterargument against security through obscurity (which was the argument being made here) is that with open source software, many more people will be looking at the source in order to find and fix these problems.

> since anyone can access the source code and change it.

Are you sure that is what they said? Change it? Not "study it"?

I don't see how anyone can just change the source code for Struts...

Share Improve this answer edited Aug 23, 2010 at 6:39

Follow

answered Aug 23, 2010 at 6:25

Thilo

262k ● 105 ● 520 ● 669

Thanks Thilo - I totally misunderstood the question then. The interviewer didnt correct me either so I thought I did fine - at the time. – newtoallthis Aug 23, 2010 at 8:12

2

A popular open-source web framework/CMS/library is less likely to have horrible bugs in it for long, since there are lots of people looking at the code, finding the bugs, and fixing them. (Note, in order for this to matter, you'll need to keep your stuff up to date.)

Now, your friend does have a tiny point -- anyone who can fix the bugs could also introduce them, if the project is run by a bunch of idiots. If they take patches from any random schmuck without looking the patches over, or don't know what they're doing in the first place, it's possible to introduce bugs into the framework. (This doesn't matter unless you update regularly.) So it's important to use one that's decently maintained by people who have a clue.

Note, all of the problems with open-source frameworks/apps apply to COTS ones as well. You just won't know about bugs in the latter til after bugtraq and other such lists publish them, as big companies like to

pretend there aren't any bugs in their software til forced to react.

Share Improve this answer

Follow

answered Aug 23, 2010 at 6:22

cHao

86.4k • 20 • 146 • 177

"popular open-source web framework/CMS/library is less likely to have horrible bugs in it for long". The important phrase here could be "for long". You may have to update quite frequently, as popular apps provide quite an attractive target, see Wordpress. – Thilo Aug 23, 2010 at 6:28

OTOH, a COTS app could easily have critical bugs lurking in it for *decades*, just waiting to be discovered by someone with the (im)proper motivation. For instance, the GDI vulnerability in Windows. Were Windows open-source, the bug wouldn't have lasted that long. :P – cHao Aug 23, 2010 at 6:36

1

a) Yes. Open source doesn't mean open binaries :) The sentence "anyone can change the source code" is simply incorrect (you can change your copy of the code, but can't edit Apache Struts code)

b) Maybe the fact that the source code is visible makes it easier to somebody to see the posible flaws it can have and exploit them. But, the same argument functions the other way: as a lot of people review the code the flaws are found faster so the code is more robust at the end.

Share Improve this answer

Follow

answered Aug 23, 2010 at 6:18

helios

13.8k 2 49 56