Checklist for Web Site Programming Vulnerabilities

Asked 16 years, 3 months ago Modified 9 years, 5 months ago

Viewed 2k times

17

Watching SO come online has been quite an education for me. I'd like to make a checklist of various vunerabilities and exploits used against web sites, and what programming techniques can be used to defend against them.

- What categories of vunerabilities?
  - crashing site
  - breaking into server
  - breaking into other people's logins
  - spam
  - sockpuppeting, meatpuppeting
  - etc...
- What kind of defensive programming techniques?
- etc...

security defensive-programming

Share

edited Jul 14, 2015 at 7:49

Improve this question

Follow

Yeldar Kurmangaliyev
34.2k ● 13 ● 63 ● 103

asked Aug 26, 2008 at 19:51

Mark Harrison
304k ● 131 ● 350 ● 489

Can someone fix the spelling mistake in the title? – mattruma Sep 18, 2008 at 14:09

9 Answers

Sorted by: Highest score (default)

12

From the Open Web Application Security Project:

1. The OWASP Top Ten vulnerabilities (pdf)
2. For a more painfully exhaustive list: Category:Vulnerability

The top ten are:

1. Cross-site scripting (XSS)
2. Injection flaws (SQL injection, script injection)
3. Malicious file execution
4. Insecure direct object reference
5. Cross-site request forgery (XSRF)
6. Information leakage and improper error handling
7. Broken authentication and session management
8. Insecure cryptographic storage

9. Insecure communications
10. Failure to restrict URL access

Share Improve this answer

Follow

answered Aug 26, 2008 at 22:20

Charles Miller

2,897 ● 1 ● 21 ● 14

6

I second the OWASP info as being a valuable resource. The following may be of interest as well, notably the attack patterns:

- CERT Top 10 Secure Coding Practices
- Common Attack Pattern Enumeration and Classification
- Attack Patterns
- Secure Programming for Linux and Unix
- A Taxonomy of Coding Errors that Affect Security
- Secure Programming with Static Analysis Presentation

Share Improve this answer

Follow

edited Aug 27, 2008 at 0:23

answered Aug 26, 2008 at 23:12

David Schlosnagle

4,323 ● 2 ● 24 ● 16

2

Obviously test every field for vulnerabilities:

- SQL - escape strings (e.g. `mysql_real_escape_string`)
- XSS
- HTML being printed from input fields (a good sign of XSS usually)
- Anything else thatis not the specific purpose that field was created for

Search for infinite loops (the only indirect thing (if a lot of people found it accidentally) that could kill a server really).

Share Improve this answer

Follow

answered Aug 26, 2008 at 19:56

Ross

47k ● 39 ● 123 ● 173

2

Some prevention techniques:

XSS

- If you take any parameters/input from the user and ever plan on outputting it, whether in a log or a web page, sanitize it (strip/escape anything resembling HTML, quotes, javascript...) If you print the current URI of a page within itself, sanitize! Even printing PHP_SELF, for example, is unsafe. Sanitize!

Reflective XSS comes mostly from unsanitized page parameters.

- If you take any input from the user and save it or print it, warn them if anything dangerous/invalid is detected and have them re-input. an IDS is good for detection (such as PHPIDS.) Then sanitize before storage/printing. Then when you print something from storage/database, sanitize again! Input -> IDS/sanitize -> store -> sanitize -> output
- use a code scanner during development to help spot potentially vulnerable code.

XSRF

- Never use GET request for destructive functionality, i.e. deleting a post. Instead, only accept POST requests. GET makes it extra easy for hackery.
- Checking the referrer to make sure the request came from your site **does not work**. It's not hard to spoof the referrer.
- Use a random hash as a token that must be present and valid in every request, and that will expire after a while. Print the token in a hidden form field and check it on the server side when the form is posted. Bad guys would have to supply the correct token in order to forge a request, and if they managed to get the real token, it would need to be before it expired.

SQL injection

- your ORM or db abstraction class should have sanitizing methods - use them, always. If you're not using an ORM or db abstraction class... you should be.

Share Improve this answer

Follow

answered Sep 16, 2008 at 8:35

Zach

24.8k ● 9 ● 45 ● 50

SQL injection

1

Share Improve this answer

Follow

answered Aug 26, 2008 at 19:52

coder1

3,515 ● 5 ● 31 ● 46

XSS (Cross Site Scripting) Attacks

1

Share Improve this answer

Follow

answered Aug 26, 2008 at 19:54

Bernie Perez

12.6k ● 13 ● 49 ● 55

1

Easy to oversee and easy to fix: the sanitizing of data received from the client side. Checking for things such as ';' can help in preventing malicious code being injected into your application.

Share Improve this answer

Follow

answered Aug 26, 2008 at 22:29

jwarzech

6,665 ● 11 ● 53 ● 74

1

G'day,

A good static analysis tool for security is FlawFinder written by David Wheeler. It does a good job looking for various security exploits,

However, it doesn't replace having a knowledgable someone read through your code. As David says on his web page, "A fool with a tool is still a fool!"

HTH.

cheers, Rob

Share Improve this answer

Follow

answered Aug 27, 2008 at 14:07

Rob Wells

37k ● 13 ● 84 ● 147

You can get good firefox addons to test multiple flaws and vulnerabilities like xss and sql injections from Security

1

[Compass](). Too bad they doesn't work on firefox 3.0. I hope that those will be updated soon.

Share Improve this answer

Follow

answered Sep 16, 2008 at 8:43

Vertigo

2,746 ● 1 ● 22 ● 24