Column

Is It All in a Day's Job?

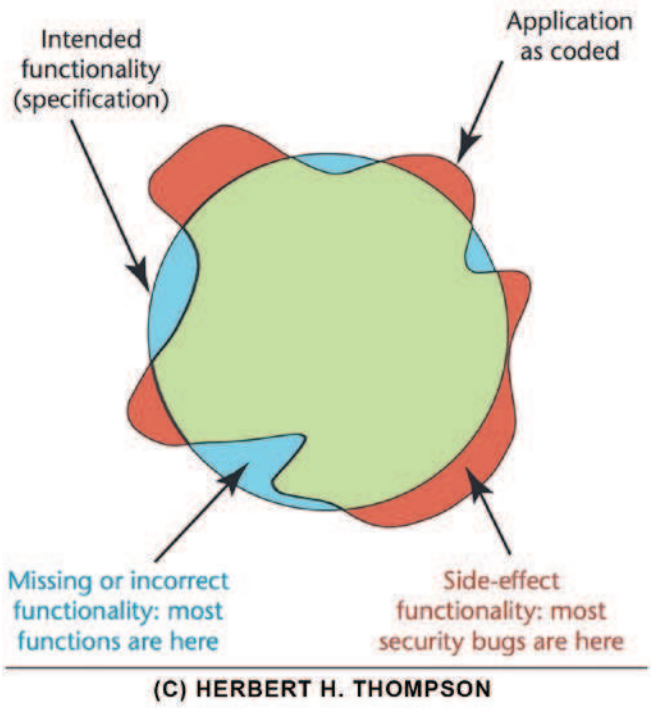
by Huib Schoots

"Testing is an infinite process of comparing the invisible to the ambiguous in order to avoid the unthinkable happening to the anonymous." - James Bach

Remember the bad performance and security of Healthcare.gov?[1]

Or the hack on Facebook?[2] In the Netherlands every year around Christmas the King's speech is published by a hacker before it is broadcast. Everybody knows stories about failing performance, usability, or security. I order loads of books and other stuff online. If web shops are slow, I click away and go to another web shop. I use my credit card online and basically hope the web shop cares about security. Only once have my credit card details been stolen and luckily it did not cost me anything.

Nowadays everything is WiFi enabled, and almost every device we use has an embedded computer. So I was not very surprised when I read about a fridge sending spam email[3] . A friend proudly showed me his new car last weekend: a fully automatic plugin hybrid with adaptive cruise control. In traffic jams he does not have to do anything, the car does all the driving. I wonder what would happen if somebody hacked his on-board WiFi and took control of his car. I do not want to give terrorists any ideas, but I wonder how many world leaders have these features in their heavy armored cars surrounded by powerfully armed bodyguards? Hacked webcams filming children getting undressed is something that was in the news in the Netherlands last week. These examples show that security is important and becoming more and more important every day.



Security testing is different!

Software testers are great in finding bugs or functionality that is supposed to be there but is not, or is different than, expected. Security testing is different. Most security vulnerabilities are built in – the software does something unspecified. Bugs like these would necessarily escape most automated testing because testers craft test cases to look for the presence of some correct behaviors and not the absence of additional behaviors[4] . I think that testing non-functionals like security, performance and usability is work for specialists. Of course, testers should be able to do

1 How secure is healthcare.gov? http://blogs.gartner.com/avivah-litan/2013/10/31/how-secure-is-healthcare-gov/

2 Behind the facebook breach and other high-profile attacks: http://www.crn.com/news/security/240148786/behind-the-facebook-breach-and-other-high-profile-attacks.htm

3 Help! My fridge is full of spam and so is my router, set-top box and console http://www.theguardian.com/technology/2014/jan/21/fridge-spam-security-phishing-campaign

4 Why Security Testing Is Hard - Herbert H. Thompson http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.4376&rep=rep1&type=pdf

some basic security testing, but only real specialists can perform excellent security testing. And hackers get better and better every day, so security testing should be done excellently. Managers often underestimate the risks involved and the skills needed to deal with these risks.

I remember an account manager I worked with who asked me to download some tools years ago. He had a client and he wanted me to go there and help them with security and performance testing.

First I thought he was joking, but he was not. The next day I went to the client and after ten minutes it was clear they needed two specialists, since the testers there tried but had not been successful in their efforts. It was not enough. Performance testing is not only about finding poor performance, but also about helping to improve that poor performance. And this is not an isolated story – I have heard stories like this in many companies.

Basic Security Testing

Last year at TestBash in Brighton, Bill Matthews gave a great talk in which he demonstrated some basic security testing. Check out the video of his talk "Context Driven Security"[5]. He talks about threat modeling in security testing, OWASP[6] and STRIDE[7]. This talk has lots of interesting stuff for learning about basic security testing. Ever thought about abuse cases? The SANS top 25 most dangerous software errors[8] was a real eye opener for me. Bill showed some basic security testing which every tester could do. His talk is great for getting started with security testing and learning more about software vulnerabilities. By modeling the threats, it will also make you aware that humans are often the weakest link in security[9].

Using the same passwords on all websites or even writing them down on sticky notes placed under the keyboard.

Security testing is easy, right? Like Bill Matthews says in his talk, it is not a problem unless it threatens the business. And, in addition, you only have to be smarter with your security then a hacker who is out there to harm you. I wish you good luck!

And do not think that it will not happen to you, because it will. My old joomla site was hacked years ago. There was not anything worth stealing – no data, no money, nothing. So I thought, why bother updating? The hackers put some malicious pages on my site and used my site for phishing, trying to get login information from blog readers. I was lucky because Google found out and sent me a message to warn me.

Oh wait. The theme of this Agile Records was "Security testing in an agile environment". So let me make this column compliant with the theme by quoting and linking to a blog post I wrote last year with the title: "What makes agile testing different?"[10] The testing itself is not so very different; it is the context in which you test that is different! I guess that also counts for security testing. But hey, I am not a specialist. So do not ask me. ■

> about the author

Huib Schoots

Huib Schoots is a tester, consultant, and people lover. He shares his passion for testing through coaching, training, and giving presentations on a variety of test subjects. With fifteen years of experience in IT and software testing, Huib is experienced in different testing roles. Curious and passionate, he is an agile and context-driven tester who attempts to read everything ever published on software testing. A member of TestNet, AST and ISST, black belt in the Miagi-Do School of software testing, and co-author of a book about the future of software testing, Huib maintains a blog on magnifiant.com and tweets as @huibschoots. He works for Improve Quality Services, a provider of consultancy and training in the field of testing.

5 Context Driven Security – Bill Matthews: http://www.ministryoftesting.com/2013/04/testbash-video-context-driven-security-bill-matthews/

6 OWASP: https://www.owasp.org/OWASP cheat sheet: https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet

7 STRIDE: http://en.wikipedia.org/wiki/STRIDE_(security)

8 SANS top 25 (Most Dangerous Software Errors): http://www.sans.org/top25-software-errors/

9 Humans: The Weakest Link In Information Security http://www.forbes.com/sites/ciocentral/2011/11/03/humans-the-weakest-link-in-information-security/

10 What makes agile testing different? http://www.huibschoots.nl/wordpress/?p=1072