Joe Rider
SE 4930
Article Review 3
1/22/12

Code analysis can be helpful in determining whether software is ready to deploy. One type of analysis related to security is static and dynamic analysis. The podcast that I listened to was an interview with Markus Schumacher from July 29th, 2011. Schumacher is a co-founder of the company VirtualForge. VirtualForge provides a code profiler for businesses that makes use of SAP's ABAP.

Schumacher was asked how he thought his analysis tool compared with others from HP or IBM. He stated that those tools provide a basic analysis of the software but since the tools were generic they could not provide very much depth. The advantage of the VirtualForge tool is that it is specific to one language so it can provide a much more thorough analysis. This is a trade off that companies have to decide whether to get a generic tool or to invest in one that is specific to their domain.

The analysis tool prioritizes the risks that it finds according to the probability and ease of exploitation. The current trend in security analysis focuses more on finding bugs rather than on fixing the bugs. The prioritization of the bugs allows developers to see what could cause the most damage so that those issues can be addressed first.

Schumacher was also asked about security patterns. He said that most of the developers that he works with understand the usefulness of patterns. However, most of them are engineers and like to write their own code and use their own design which leads to "new" security solutions being developed from time to time. Schumacher also noted that patterns do not necessarily have to be applied at the design level. Patterns can be used at the architectural or code level.

Since Schumacher is the co-founder of a start-up company, he discussed working for a startup versus a large company. Although this is not strictly related to security, it is relevant for a new developer looking for their first job. Schumacher said that the best thing about working for a startup is that there is more freedom to do what needs to be done. Developers have more responsibilities as they often must be developers and testers. He said that the best part about working for a large company is that the domain is larger.

This is a highly relevant topic for future development projects. I may never work with ABAP but the overall lessons can still be applied. Code analysis tools are a good way to automate the analysis process and at least get an initial idea of what may need to be fixed in the product. It was good to hear that some of the topics that are being discussed in lecture are being used by professionals. One of the patterns that Schumacher discussed was the use of trust boundaries. This is a pattern that was discussed and is relatively easy to integrate into any system that has multiple components. The other main point was the prioritization of defects. This allows developers to justify a decision to management to delay releasing a product in order to fix the most critical security bugs.