Dan Bednar
Article Review 3
01/22/12
Silver Bullet #60 with Neil Daswani

This podcast was an interview with Neil Daswani. Neil is the CTO and founder of Dasiant, and in the past was part of Google's Software Security group. He has also co-founded Stanford's center for software security certification program.

Neil Starts off his talk almost immediately talking about security in depth. He expanded on this slightly later but went off to say that it was a key part in keeping software secure. When he went back to this he mentioned that it's impossible to prevent every attack. He also mentioned that it would be difficult to test for everything as well. However there is something a programmer can do, code for recovery. If a program can detect, and recover from an attack without much time-loss, it can be considered successful.

Neil also talked about the importance of planning for the security. He mentioned looking at security and threat models as well as mitigations right from the get go when starting a project. He continues to mention that these should be reviewed during implementation and testing and not just at the product review. This was a new, concept to me but it should have been an obvious one. It only makes sense to ensure what I code is secure and matches the security requirements.

A large portion of the podcast was spent talking about Google's safe search API, also known as Google's blacklist. One of the largest points of the API is to help stop drive-by downloads. Neil mention's this is a response to malicious users using adds to trigger downloads that adds contain. Neil mentions how many companies have learned the hard way how adds can trash websites. Even the London Stock Exchange has gotten temporarily black-listed due to malicious adds.

Neil goes on to state that now adds are being run through to ensure they are clean. In response to the malicious code, add companies are running their databases through what is essentially an anti-virus. This program verifies that code will not run anything malicious in the background.

One thing Neil mentioned was doing more penetration testing while in the testing phase. This is a logical place to insert penetration (pen) testing. I've always pictured pen testing being one of the final things that's done once an entire project is integrated. But why not spend an extra hour or two testing with pen tests? It would make it easier to find errors and fix them properly when tested individually.

He mentioned the security through depth here as well. Automated pen tests can help uncover flaws and risks that the manual tests cannot find, or the testers do not think to find. They can also attempt to drill down through multiple layers in the tests. The justification was that the attackers will be using automated tools to make things quick and dirty so they should too.

One thing that either Neil didn't mention or I failed to catch was testing and security reviews during the review phase itself. I'm assuming it's implied but it would have been nice to hear his opinion. How vital does he feel the tests are during the review phase as opposed to the coding and testing phases of the project?

I thought it was interesting to hear his perspective on how the security process can be taken and applied to the real world. It was interesting to see how Google's safe browsing API used these principles and how it can protect people from drive-by downloads. However, I feel it wasn't complete. I feel like he missed the after-development, bug fixing part of a projects life cycle. How can a project team ensure their original security goals are not compromised by field-found bugs? Does he think companies should keep automated tests for older product versions to do this, or is there something else that could be done without spending multiple man-hours?