Fred Cohen & Associates - Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Fault modeling, the scientific method, and thinking out of the box

I have been identified by many folks as someone who “thinks outside the box”. I personally reject this notion and sometimes reply that I just think in a different box. We all have limitations and these limitations form the box we think and act within. The box is formed by a combination of nature and nurture, and in my case, part of my graduate level nurturing was being educated in the ways of fault tolerant computing by folks that were there when the field was just developing. This is a key to understandings that I have of how to make progress on many of the seemingly unsolvable problems we face in the information protection arena. So I figured I would share the key and see if we can, together, unlock a lot of the boxes out there.

In the field of fault tolerant computing, there was a breakthrough many years ago when the leaders of the field make an effort to understand and create an epistemology for their field. I am sure that this has happened in many other fields, including the philosophy of science, which I studied a bit as a gradual student as well. But it is the lesson from fault tolerant computing that I am discussing today, and I am anxious to find out from the real founders of that field just how wrong I am about all of this, now that I am sticking my neck out and calling them brilliant. The breakthrough was that they gave up on trying to be perfect and decided that they needed to create and apply fault models knowing that those models were imperfect.

I bet you were thinking it was going to be something really profound – didn't you? Actually, it is really profound – it just seems pretty mundane.

Once you decide to stop trying for perfection and start trying to model something useful, you might do any number of things, but one of the things I see done least often is the other half of the brilliance of fault tolerant computing and of the philosophy of science. That is, modeling faults instead of successes. This is the notion of refutation rather than confirmation that is so core to scientific progress. As the theory goes, from the philosophy of science, a theory can be refuted by a single example – but any number of confirmations cannot prove it to be true. Actually, that's a simplification. A scientific theory about an in finite set is more precise, and it deals with observable phenomena (science) rather than purely theoretical or mathematical notions that can indeed be proven by derivation or induction from assumptions. And that is the breakthrough that I think we need in information protection – not just here and there as we are starting to see it, but almost everywhere.

Here are some examples:

- How do we analyze forensic evidence and analysis?
  - Since we can't tell if it right for real-world cases because we can never get enough of the necessary information to really prove it, we apply a fault model. In this case, there are “make and miss” faults in the forensics process that results in false positives and false negatives. Specifically, the forensic process can be broken down into various steps taking whatever model you choose to use, and for each of those steps, you can make or miss content, context, meaning, process,

relationships, ordering, times, locations, consistency, and corroboration, and do so by accident or intent. I went through these possibilities in some detail in my recent book on "Challenges to Digital Forensic Evidence".

- How do we determine if a business model is suitable to a business?
  - Because we cannot derive a suitability criteria from some notion of the business, and every business is different, a reasonable alternative is to list the things we find in many or most businesses and start from there. If we haven't considered any of them, we likely have a fault in our analysis. Of course we will still be missing things, but we won't be missing these things. I did this analysis briefly in my recent book "Enterprise Information Protection".
- How do we determine if the security inventory is adequate?
  - If you don't have one, it's inadequate... but beyond that, I recently saw a talk where it was explained – and credibly – that about 70% of externally driven losses from enterprise attacks were losses from targets that the enterprise didn't know about. In other words, they didn't have the thing that was successfully attacked in the inventory. The fault model here is fairly simple. If it's not in the inventory, you are likely not properly protecting it, and of course, the inventory has to be accurate or you don't have it in the inventory.
- How do we determine if our anomaly detection will work for us?
  - The whole notion of anomaly detection is based on the fundamental principle of knowing what is supposed to be and searching for deviations. In other words, anomaly detection is, in practice, the creation of a fault model for whatever the detectors are supposed to be examining. In this case, it is the modeling of the faults that determines the utility of the detection mechanism, and that means that we need a meta-model of anomalies in order to identify the modeling faults. I haven't created one of these – it's still on my list.
- How do we detect errors in code for software we deploy?
  - This follows very directly from the fault tolerant computing models associated with hardware. We create fault models, such as the buffer overflow or null pointer, and develop analysis for it. Only the breakthrough here will be in the area of composition. While we are eating away at lots of fairly obvious fault types in software with static and dynamic analysis tools, what we lack is a composition model that shows how faults arise from the interaction between components within a composite in an environment. But that's a story for another day...

You can add to the list as easily as identifying an issue, stating it in the proper context to apply fault modeling, and thinking through it fairly quickly. It's that simple. And it has worked for me for years. Maybe it can work for you.