Nathan Dykstra
SE 4930
Article Review 3
January 24, 2012

Article Choice

For this review, I chose to listen to the Silver Bullet podcast show. It was episode 41 and aired August 21, 2009. Gary, the host, interviews Fred Schneider who is the Samuel B. Eckert Professor of Computer Science at Cornell University and author of Trust in Cyberspace.

What Was Discussed

Gary starts by discussing Fred's work on mission critical, high integrity settings and asks what the relationship between security and reliability. Fred says that they are really closely tied. He points out that the difference is the cause: reliability is typically about random events that one can attribute to nature (physical processes, developers making programming mistakes on large systems) and security is attributed to malevolence.

Fred talks about his work on distributed systems and how the main concern was if one of these systems fail. He mentions that not much focus was put on redundancy and these failures were quite significant. Fred points out that many of the security concerns during this time was ensuring your program wasn't interfered with by any other program running on a timeshare system. This is because the internet wasn't widespread and no true hackers really existed.

He goes on to talk about system failures in a distributed architecture. Most people view the replication as a fail safe, however Fred points out that this really only applies if the failures are independent. He uses the example of redundant systems all using the same backup power supply. These would all have one related point of failure and nullify the benefits of a replicated system. This is also true if vulnerabilities exist in one system; this vulnerability will be replicated in all other systems creating more openings for attackers.

Fred mentions how one can build diversified replicas from the ground up to really ensure a redundant yet secure system. He mentions how so people somewhat emulate this in the real world by generating different code through randomizing the storage layout, using different compilers, or compiling to different instructions. This isn't truly independent but is more practical than creating totally independent and diverse programs.

Gary and Fred discuss some topics we've already covered in class. Specifically, the differences between bugs and flaws. They also talk about trust issues within software like the trust boundaries we worked with. Fred talks about 'enclaves' that are in the trusted area. He makes the observation that if one of the systems within this enclave is breached, all systems are breached. And these other systems may not be aware because any requests from this infected system are coming from within the enclave and are trusted peers.

One of the main points Fred talks about later is the different ways of securing software. He reminisces about his early career when he would formally prove algorithms as a preventative measure. Few places perform this nowadays. He then talks about accountability and draws the analogy of police solving a crime and convicting someone. In software, this is basically attributing actions to the principles, typically a person. This can be problematic in the real world as many attacks are performed as some actor that has access to the system by an external actor that is directing the attack.

What I Learned

Much of what they talked about we have already covered in our class such as bugs

versus flaws and trust boundaries. However, it was interesting to hear Fred talk about how security was a slightly different concept before the Internet. Specifically, how security focused on preventing other programs on a time share interfere with yours rather than malevolent threats from outside the network.

I honestly didn't gather much from his talk about replicating systems either. The examples of single points of failure and security flaws being replicated in redundant systems seems self evident why the are bad practice. Unfortunately, Fred did not go into much detail about the processes involved in creating diverse replicate systems from the ground up as this is something I find intriguing.

How It Will Help

It was interesting to hear the history of how security changed over the years, but as I mentioned, most of the topics were either covered in class or self explanatory. This may be due to Fred's generalizations and lack of detail, however.

Questions

None.