All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Aurora and why it doesn't really matter

Aurora is the name given to a vulnerability demonstrated in an experiment performed in or about 2007 by the US Department of Homeland Security (DHS) and subsequently released in various ways until a few weeks ago, when most of the full history of the experiment itself was released in response to a freedom of information act (FOIA) request. Former DHS employee Perry Pederson has publicly indicated that he was the author of most of the content released, and that he led, at an administrative/management level, the team that did this experiment. The experiment is often shown as a large rotating machine shaking with increasing violence until it destroys itself. The claim is that this was caused by cybernetic manipulation of the control mechanisms from afar, and this has been confirmed publicly.

What's the current debate about?

There has since been some non-trivial amount of debate regarding the tradeoffs between information sharing and secrecy, particularly with regard to the releases of information about Aurora. In the wake of these events, I thought I would put some context around them, in particular with respect to the Aurora demonstration, about which I have no direct knowledge, but a great deal of indirect and possibly inaccurate knowledge, and the related releases of information.

The current debate surrounds the time it took to reveal the details of the experiment so that potentially affected organizations could respond to them, and the nature of disclosures over time, both in this case and in the more general case. In essence, this goes to the issue of full disclosure of vulnerabilities vs. limited disclosure vs. non-disclosure, issues of classification, and issues of equities. While some are unhappy that full disclosure wasn't made immediately, others might reasonably contend that the disclosures recently made went too far. My view on this later... but you may have a hint from the title of the article.

Delusions

We all live in our delusions. Without them we would likely be very unhappy. We feel we are somehow important, we think we are handsome or beautiful, we think we are athletic, knowledgeable, skilled, etc. Until we meet someone who is just plain better... and then we may resent them... But that's just being human. Management has all the same issues. And that is, in my view, the nature of the challenge we face.

Some people think that keeping things secret is helpful, while others thing it is damaging. And they may each be right from their perspective about the same thing in the same circumstance. But this is not the central delusion surrounding this issue in Aurora disclosures. The central delusions today are:

- That disclosure will cause anything to get done by the defense
- That anything disclosed would be new knowledge to reasonably knowledgeable people

I assert that both nothing new other than the scientific demonstration of a well known principal was involved in the experiment, and that full disclosure would not have gotten anything fixed.

Disclosure doesn’t solve the problem of mitigation

Some people claim that in order to fix a problem you have to allow knowledge of the problem to disseminate to the people that can fix it. I don't agree. This may be an efficient way to do it in some cases, but it may not be. Let me just ask:

> How many patches have you applied without the full detailed knowledge of the underlying basis?
>
> Does your organization require that before anything is patched, all of the team members involved with the relevant systems and interdependent systems be notified, trained, aware, etc.?
>
> Or do you get a patch from a vendor and apply it during a maintenance window to a test system, verify that nothing appears to break, then apply it over time to operational environments?
>
> Or do you just apply the patches and hope?

In my experience, almost nobody does the dissemination of the knowledge of the details of each vulnerability and how to fix it at the detailed level before applying patches. Most organizations don't even do do the change control approach. And yet most such fixes work well. So in summary, you don't need to know everything about a problem in order to fix it.

Another interesting claim I heard is that you can't know who all the people are that can fix a problem, so by telling more people, you are more likely to reach the people who can fix it. I agree that this is usually, but not always, the case.

I know all the folks who have to fix things in my infrastructure. And in some cases, there are small communities involved. But then there is the question of who you want to fix it. If you are the US government, apparently you don't want everyone to fix everything. Because the things that are broken are part of your offensive capability. So part of the apparent plan (and I have no actual knowledge to back any of this up) is to only have select providers fix specific things so the US is a bit safer and the rest of the world can still be attacked by US intelligence agencies and military. This is called the equities issue.

I am not supporting or rejecting any particular viewpoint here, but I think it is reasonable that depending on who you work for and what your goals are, you might decide that different people should know different things.

And there is another important point to be made. Wider disclosure exposes the details of the problem to more of the people who might want to use it against you. Full open disclosure opens up the potential that attackers could exploit the problem before defenders could fix it. Indeed attackers are generally more willing to act than defenders, at least today in this arena.

And I don't think the challenge is necessarily a technical one. In my experience, even when you know to fix a potential vulnerability, getting it fixed may take an unlimited amount of time. The real advantage to the attackers is that they may act immediately and need no permission to do so. Defenders on the other hand are often constrained organizationally or otherwise strapped for resources. That's why the announcement of new vulnerabilities followed by the patch cycle still leaves a significant percentage of all affected systems vulnerable 6 months, a year, and several years later. Attackers use the knowledge while the defenders will not. And that is a good reason to restrict the information to the defenders – to give them time.

However, some problems with this approach include (1) it doesn't matter how long you give them, they still may not get it done, (2) the attackers will eventually find out anyway and likely sooner than you think, and (3) a secret is of little use unless you can share it with the right set of people when appropriate.

There was nothing new in Aurora as currently disclosed

I now harken back to the deeper issue at hand - the vulnerability now identified as Aurora. Note that this particular example of a larger class of vulnerabilities has been known for a long time and is a relatively simple extension of a long-known principal in all of engineering - positive feedback. All educated engineers know that unconstrained positive feedback grows a wave till it exceeds the capacity of the media, and then bad things happen. We even have engineering equations to figure out how to reduce or eliminate ringing and over-damp mechanisms as appropriate. The bridge collapse movie has been with us for many years.

The particular issues in Aurora were identified long before the dissemination of the video, and yet it wasn't fixed. The issue of positive feedback in systems has been known for a very long time - certainly from before I was born, and yet we still don't design systems controls so as to mitigate this.

Don't get me wrong. It's not that the engineers don't know about it. Cybernetic (i.e. control) systems are often if not usually designed with this in mind. But the use of automatic control systems in the digital era, as many things in the digital era, failed to properly consider the potential of these systems to go awry in ways not existent in analog systems. In analog systems, fixed value components (e.g., resistors and capacitors and inductors), aren't alterable from over the Internet by a computer program. The problem we have here is that we don't have the same engineering disciplines and quality of mechanism controls that we used to have.

What to do about it

This is not to say we should be going backwards. Rather, we should be going forwards with the recognition that essentially all current digital designs in widespread use are far less dependable than almost any analog system of 50 years ago. We have made a giant leap backwards in surety while making a giant leap forward in controllability and function. What we need to do now is get all of our engineers, operators, etc. up to speed on this and have them start rethinking and redesigning all the crap they implemented over the last 15 years to make it as high surety as older analog technology while maintaining the advantages of the newer technology. To do this, management has to recognize that they were given promises of efficiency without the right price tag, and has to put up the money to pay for the value they got - or at least stop being deluded about this particular thing.

Summary

Which brings me back to my central point. We all live in our delusions. The underlying reason that cybersecurity is so poorly done is **not** that there is a crying need for more research and development, or that it is impossible to secure these systems. It is that we fail as a society to apply the lessons of the past. Indeed we fail to even review the past and seek to understand how it applies to the present and future. It's easier to say something like “Who could have ever known?” But it's better to do your homework and be a professional in your field. And that's why Aurora doesn't really matter. Because we still haven't learned from it.

