All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Bad decision-making OR Making bad decisions?

There is a difference. Everyone makes decisions that come out bad. That's what I call making bad decisions. But bad decision-making is when your system of making decisions is poorly suited to the decisions you have to make. That tends to produce bad decisions because of process errors, and becomes a systematic path to making bad decisions.

An example

In cyber-security we see this all the time. Someone decides that we (as a company) should use this method to do this function. Perhaps the decision is to use Windows for processing payments. Good decision? Bad decision? I have no idea. It depends on the situation.

How did they make the decision? I (as a company) have no idea. And that's the problem.

What's wrong with that?

The fundamental problem from a cyber-security standpoint is that business decisions have consequences, and we need to understand the consequences in order to make sure that the company isn't whacked.[1]

- The nature of the problem is that it's not so obvious whether a decision might cause these sorts of bad things to happen.
  - So the way we make better decisions is to check the proposed decision to see whether it will have some (hopefully unintended) bad consequences.
    - But how do we do that?

Workflows

The general idea is to check what we are going to do against what we are doing and why we are doing it that way to make sure we don't do something stupid.

- But it rarely works that way because people making decisions don't check, or even think about the possible negative implications of what they are doing.
  - So workflows provide a method to assure that the proper (or at least some) checks are made before doing stupid (or smart) things.
    - But that slows down the process of doing things.
      - So people decide to ask for forgiveness or hope they don't get caught.
        - Which is also how most things that get done actually get done.
          - Because lots of people try to stop you from getting anything done.
            - At least in any substantial enterprise.

So care must be taken to assure that process isn't interfering with progress.

1 A technical term that means killed / destroyed / seriously damaged / or something like that...

An example

But mostly it's something else. People simply do not want to think about things they do in the larger context, and when they do think about something, they get a good idea and proceed with it. It's hard to think through everything you do all the time, and it tends to limit creativity.

We recently performed a pathfinder (a process to identify the current situation regarding cyber-security and identify a path forward) for a company.

- In the process we identified that they held no personally identifying information (PII)
  - That translated into a series of decisions relating to protection that eliminated potential costs, effort, attention, etc.
    - So an architecture was identified to provide reasonable and prudent protection on that basis.
- A few weeks later, in a discussion about other issues, a casual point was made about how some information was going to be used for some purpose.
  - As it turns out, that information was PII.
    - And because there was no process in place yet, this was detected by accident.

The people making this decision were participants in the previous pathfinder only a few weeks earlier. And yet they didn't remember that they had an assumption about not having any PII. It just never occurred to them to think about this, or they failed to get it into their brains, or perhaps they forgot their previous decision, or whatever.

- So I pointed it out – by showing them the assumption they made.
  - Because I happened to remember it
    - And because we document such things and are readily able to look them up.

But that's no way to run a rodeo[2]

The thing is, finding such stuff should not be the accidental coincidence of my being in your meeting when you mention something, and me happening to remember what we talked about a few weeks ago.

If you are going to run a substantial business successfully, you have to get systematic about things pretty quickly. That means documenting things, putting regular processes in place, and so forth. And it means that every time you do something new, you need to make sure it doesn't screw up something old.

But innovation!

Yep. That's a problem. If you want to innovate, you need to be able to move forward quickly. And that means one of two things;

(1) you better know enough to find and fix the problems before you encounter them, or

(2) you better disaggregate your risks.

2 I was going to say "this isn't my first rodeo", so my apologies for the quaint expression. I guess I had bulls… and other things on my mind.

Huh? Disaggregate my risks?

Right… pull one little thread and the whole cloth comes undone. Or to be more accurate, the opposite of that. Risks aggregate when things are interconnected. One thing goes wrong and it cascades into a major problem out of control. How do we prevent this?

In the parachute business this is called ripstop. It is a method of weaving that prevents tears from spreading. That way, the risk of a tear is limited to not cause the whole parachute to fail. The fabric is a bit more expensive, but would you decide to save a few dollars per parachute by not using ripstop? I suspect regulations would stop you. If you were following regulations.

In essence, a way to avoid a bad decision becoming a disaster is to limit the consequences to a set of event sequences that are not unnecessarily debilitating. You can do that be removing the aggregation of risks created by interdependencies, which then is sometimes (often today) called supply chain risk. Only it works for internal as well as external supply chains.

Security Orchestration and Dev Ops

Of course I couldn't leave you without adding these two terms to the pot I am stirring (or the hornet's nest I am opening up – or is it Pandora's box?). Plus, the addition of those words means that this article is more likely to be found by search engines… but I digress.

Just kidding...

If I am going to systematically disaggregate risks, that means I am likely going to have to know what they are and be able to identify them. Which returns me to my original issue. In cybersecurity ignorance is not bliss, it's suicide.[3] We need systematic ways to make good decisions, which is to say, good decision making.

Security orchestration (SO) is an example of how we can largely automate the complex orchestra of decision-making and execution in an information technology environment. Development and operations (DevOps) are the medium that SO operates on. SO DevOps is all about automating real-time decision making and execution, the decision-making largely taking place in the Dev part but being executed in the Ops part. The resulting symphony of activity that takes place in real-time to deploy resources, provision services, authorize activities, and trace and analyze performance, enables a control system that uses feedback to automatically manage operations of a type and at a scale that were previously infeasible. This is the logical extension of identity management integration that started in the 1990s.

Governance (the G in GRC™)

Unfortunately, or perhaps fortunately, there is also human decision-making involved in actually running a company. And we can (should?) not (yet) automate this function. This is called governance. It is the missing G in Governance, Risk, and Compliance. And the problem with good governance is that it's hard and complicated and not fully automatable. A systematic approach to governance that involves a systematic approach to making decisions and checking them against their implications is the way out. Plan, do, check, act. [Demming]

Conclusions

Making good decisions systematically calls for good decision-making.

3 You should perhaps reference F. Cohen, "Introductory Information Protection", 1987-1990 where this originated. http://all.net/books/IP/Introduction.html

