All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Cybersecurity From Scratch – Part 4: Impulse engines (a few bursts of energy)

Every once in a while we get to create a cyber-security program from scratch. …

The COVID vaccine has produced a seemingly minor disruption. The IT Director reports having been sick for some time related to the 2nd shot, and this is responsible for a slowdown in progress against goals. In a frank discussion, I indicated to the CEO that I could execute on the most critical tasks but that this would be more expensive (although it would happen in fewer hours and right away it would be at my consulting rate) and would not keep the IT director up to date and build their capabilities for the path forward. Thus the eternal tradeoff between individual effort and a corporate solution. More about this at another time...

Time marches on

Building the 120 or so decisions behind cyber-security program governance is not something done immediately or lightly. Even in a small company, resource constraints and decision-making speed are limited, especially since you have to run the business. To make 120 governance decisions from scratch over 6 months means 20 decisions a month, or about 5 decisions per week. These decisions have implications too… Once you decide to do something and codify it in corporate documents, you exposure from not doing it increases because you are now responsible for doing it. On the other hand, simply not making the decision, you are responsible for ignoring something you reasonably should know about. The solution to this bind is not ignorance. It is sound decision-making and an approach to constant improvement. With limited resources, you cannot immediately do everything, but you can reasonably prioritize and and prudently do this higher priority (urgent and important) things sooner, planning to do other things as you move along. Thus the approach to getting to Defined maturity in 6 months is aggressive but doable.

Documenting decisions

One of the key areas you cannot really wait on is formalizing the documentation of decisions and the response to risk. If you know of a risk and fail to document and respond to it, you start to run into the lack of prudence, since any reasonable person would, knowing of something that can go wrong, decide what to do about it in a reasonable time frame, and codifying what you do creates a contemporaneous normal business record that can be reasonably relied upon in the future. Some executives seem to think that by putting it in writing, you create liability, presumably under the notion of “plausible deniability”. In other words, I can lie about it or not remember it properly later and get away with it. The problem is that someone else will remember it differently and then we have a battle of memories and lies. The side with the contemporaneous normal business record will win this battle in most (legal and other) cases.

The final codification of a decision is what I call a Decision Document. It codifies what the decision is (was), its basis, approval(s), effective date, re-visitation date (longevity), and other relevant information about what it applies to. For cyber-security governance decisions from the Standards of Practice, there is reasonable structure to do this, so I added a form to JDM to allow our existing system of records to act as a temporary repository for this client.

What it looks like

It’s not pretty, but then that’s not its purpose. Here is a sample I produced for a phony decision to give you a sense of it:

Risk Management Decision Document For COMPANY

Saved as of 2021-04-27T14-11-04 by Identifier (Identifier)

Decision Identifier: 2021-04-26T16-40-17-29752	Issue: ◦ Knowledge: How is the knowledge program integrated with cyber security?

Description of the decision:	Immediate training of all workers on backups and upcoming changes to IT operations.

Basis of the decision:	SoP review of a potential future state informed by Urgent need to get backups from all users then conversion to new systems.

Approved by: FC	**Approved date:** Sat 2021-05-01	**Effective date:** Mon 2021-05-03	**Revisitation date:** Wed 2021-11-03

Priority: Urgent	**Management:** Governance	**Oversight:** CEO	**Risk Area:** Operational	**Risk Response:** Mitigate	**Interdependencies:** N/A	**Area:** Training

Lifecycle: People	**Process:** Prevent	**Element:** N/A	**Mechanism:** Behavior	**Control:** N/A	**Objectives:** Use control, Transparency

It has the key elements identified above, in this case linked directly to a decision from the SoP (the knowledge program) and indicates a decision to perform immediate training based on the SoP and related information. It is urgent priority, a governance matter, overseen by the CEO, and mitigating an identified risk, etc. Note that this particular form is not internally consistent with the actual decision, just a sample. The “Saved as of … “ at the top is the record made by the system of the identifier used to save (approve) this document, and by putting your initials in the Approved by box, you are specifically acknowledging that, logged in as Identifier and acting as the second Identifier, you agreed to this. As a system of records, every change you make is retained, so as/if you change the fields in the form and save, you codify the changes you made to the decision. If you make some in the approval process, it will be clear who did it when, and of course that you were changing your mind – or at least changing the form.

At this point, there were about 15 decisions already made but not documented, only noted in Gigs and potentially producing updates in GWiz™. By the weekly Pathkeeper meeting these were all entered into Decision Documents and made formal by the CEO, at which point 15 things to get done were now completed until the re-visitation dates, and queued up in Gigs for revisiting on those dates (typically 6 months out for Medium risk Defined maturity). In practice it takes about 4 minutes to fill in the form for a decision made. Another minute to approve.

New decisions

The CEO sent an email to all parties identifying key awareness issues relating to the cyber-security program, and decided that a good way to document things for the team was to send out copies of this series of articles… at which point the association between these articles and the company became internally clear, and the need to keep everyone informed of their confidentiality requirements became particularly important.

WORKERS: THIS IS NOTICE TO NOT TELL OTHERS THAT THESE ARTICLES ARE RELATED TO YOUR COMPANY

Of course, the dis-association of my articles from the companies they are about is intentional in that I can either list clients or provide useful information, so for about 50 years I have decided in favor of the useful information over the listing of clients. I also make minor changes to facts here and there in these articles to obfuscate company details without meaningfully changing the utility to the reader. So be warned – we are now in uncharted territory.

However…

The company also just got a major commitment from a major investor and channel partner. As a result, while the security process is continuing, the CEO is very busy dealing with the new issues and new commitments that come with it. The value of the company is likely to double very soon as a result of this investment and the connections they bring into the market, and that means new people have to be brought on. They need a system operating ASAP.

But in the meanwhile, the IT Director is having problems as likely side effects of a COVID 2nd shot and is now going in for additional medical care. As a company with only 8-12 workers (including us) and now soon to grow to more than 30 (within perhaps 45 days if things go quickly), they know they need to hire more help to get the job of supporting IT and protection within the company done.

As if that were not enough of a challenge, their bank, presumably in keeping with Know Your Customer (KYC) regulations, has decided that they are bringing too much money from investors. So, the bank refused to allow the use of a received incoming wire transfer (for now and who knows how long). Naturally, they did this just before a payroll was about to be made, and then called the CEO to tell him that his payroll could not go through because there was not enough money in the account. I have had a similar experience with PayPal, which decided at one point that my company was growing too quickly and stopped payment on incoming monies, putting the money on hold for an asserted 30 days. But they had no problem in sending outbound payments… of course. The CEO is, of course complaining to the bank, and looking to switch banks. This is a typical challenge for early stage companies rapidly growing and changing. If all else fails and people are not patient, the CEO can always (I say always, but usually personal funds from startup CEOs are not big enough for that sort of instant out of pocket expense) send them the money from personal funds, making these loans to the company, to be repaid immediately upon receipt of the incoming deposit. This also creates a problem with arms length issues, but you need to keep the company going and that’s just how it goes. Take the risk to get the reward. I typically keep copies of the last payroll all the time so I can just do the same thing again, and then fix it in any subsequent payroll... I have even advised this for large enterprises in some cases.

A key strategic decision starting to come together

The key strategic decision at this point that will hopefully get the company from here to there in a reasonably safe and efficient way is the decision to deploy company-owned and supervised Apple (likely Air) computers to the end points, provide managed Windows virtual desktops on centralized (Google) cloud servers, support management of the endpoints systems through a 3rd party (likely IBM) provider using a consulting firm to do initial deployment of the systems and provide ongoing support for that aspect of operations, and to use Google multi-factor authentication and federated identity management to do overall control of services. Backups will be automated and multiple cloud locations used, along with downloads to a local copy by the CEO (who is also a major shareholder) for safekeeping. All of the disks and communications will be encrypted, and various other residual risks will be accepted for 6 months as this plan gets into place, and as we walk through another 100 or so decisions that will likely all operate reasonably well within this framework for the meanwhile.

Approvals:

Using the new mechanisms for documentation of decision approval, the CEO approved Scope (on behalf of owners), Protection Model, Security Consultants, Outsource Things, Maturity, Risk Definition, Location, Form of Duties, Knowledge Program, Dependencies, Duties Prioritized, Mobility, Org Structure, and Outsource People issues from the previous weeks. In essence the decisions were copied into JDM Risk Decisions, categorized in terms of aspects of the protection model, entered into JDM for approval with verbal approval dates, and formally approved by the CEO by saving each document logged in as their identifier acting as the security governance role. This final approval took a few minutes once started. Once approved, the decisions were marked as “ignore” in the sense of no longer appearing on the list of things to be reviewed by the CEO, and Gigs was used to queue them up for re-visitation at the approved re-visitation date. As they come up again and are reviewed, they will again be configured for visibility by the CEO so they can be again marked for the next period and changed to reflect new decisions likely after Defined maturity is reached. The next step will be Managed… we suspect.

Execution support

Separation of duties demands a separation between “specify”, “execute”, and “verify”, even for a company of this relatively small size. It may seem unusual, but we resist becoming an execution partner for out clients in order to retain this separation. That’s the reason the CISO is not the same as the temporary CIO (me), and the reason I don;t want to get engaged in execution as the head of their advisory board. But in order to help client is critical situations, I sometimes step over the line and help in execution, constantly reminding them of the issue and trying to get back out of this role ASAP. Our allowing them to use JDM for risk management decision-making support is an example of such a thing, where we are keeping copies of their corporate records for a limited time period to facilitate action in a time of need.

Conclusions

With the expected fits and starts of any new program, things are running at good speed, and decision-making is reasonable and prudent as far as we can tell. Execution for a small company planning to double or more in size in 6-8 months has become a necessity in order to maintain adequate control as new workers are brought on board.

Changes in the Tools

Of course the tools evolve to meet the changing needs of the clients. I don't charge clients for the time I spend changing the tools, because that's part of the value I add to my tools as I work with clients. I leverage the tool changes for multiple clients over time. But it's part of the work I have to do to support clients, even as I tell them to not become dependent on my tools for their companies. This may change as the volume increases, but maintaining a 24x7 SaaS platform is not my goal as a consulting firm doing governance consulting. It is, however, the goal of one of my other companies that ultimately will take over such maintenance functions as/if we start to scale in the governance issues beyond a few tens of clients at a time.

I have a list of a seemingly unlimited number of things I want my tools to do for me, and I use them with clients as I have since the 1980s. They have evolved over the years, and every time I find my self doing something automatable the 3rd time, I wonder why I didn't automate it the 2nd time I did it. This week, I finally did some things I have delayed for years to make it easier for clients. **Here is more than you ever wanted to know...**

I added an identity-dependent library capability to allow identities to selectively add specific decisions.

- Whereas I used to add such decisions for clients and have access to the 2429 current JDM decision elements codified into sets of forms in groups up to a few hundred, I now provide links between identifiers and sets of decisions that each identifier can use (in the proper mode). Most such identifiers have access to a few forms that they need for their tasks, and as much of the content and mechanism is trade secret, I need to reasonably control sharing and access based on need to use and existing agreements with clients relating to such trade secrets.

I am adding a search and select capability to JDM that already exists in GWiz™ and Gigs.

- This was in the planning, but because JDM has historically been used for a small number of extensive forms, search was never required for anyone but me as an administrator. However, now that clients are able to add their own Decision Documents and other similar things from their identifiers' libraries, they will need to be able to find subsets of decision document and other similar things out of lists of hundreds of things, so I added this in anticipation of their future need.

I added the mode selector to JDM (already present in other applications) so that Novice users need not see all of the options till they get used to the tool. The mode selector includes Novice, Normal, Look and Feel, Advanced, Expert, and Admin modes.

- In Novice mode it's hard to make any mistakes that would make anything visible or not visible (for example), or that would instantly add something that would create a record.
- In Normal mode, you can do most things most folks do most of the time.
- In Look and Feel mode, you can change the look and feel of the display for your use. This includes colors, rounded corners, font types and sizes, and similar things that users may desire for personal preference.
- In Advanced mode, you can perform almost all functions on a single item (e.g., a form for JDM, an item in a sieve for Gigs, etc.). This is where I added the capability for

library access. Thus as users set themselves to the Advanced level, they can add their own (where authorized) forms from their libraries.

- In Expert mode you can do things in groups, such as change the settings of all selected items, for example, make all Decision Documents more than 180 days old temporarily invisible to the user, set documents more than a year old in an archive, etc.
- In Admin mode you can turn on and off debugging and set the debugging level. Debugging, at the highest level of volume, produces internal details of values of tables and inputs and outputs, hash table content, array elements, and generally makes the system completely unusable except for the purpose of identifying some internal variable name associated with some problem in the system.
- Of course not all of these modes are available to all identifiers, and depending on the applications you can access, different modes may be available to you.

I improved the ActAs capability and did (am doing) a restructuring of identifiers for select clients who have grown to the need. This goes to the authorization mechanisms of the tools, which are a variation on roles and rules. Here is more than you ever wanted to know...

- In my tool sets, identity management operates differently than in most systems. In essence, identifiers are authenticated by authenticators, and permit those using the identifiers to use applications and access items based on knowledge of an internal state.
    - All activities are always in a state. That state is either explicit or implicit. An implicit state is associated with open external access. You can see anything in the public view if you have it's URL, even though there may not be a path to find the URL without having had access to an implicit state to provide it to you.
    - In an explicit state, the state is presented to the system and produces an output. The output displays what looks like an application interface that is used to provide input and produce a new state with it's associated output. States never recur, so you can only move forward from current state to next state, and if you don't have a valid current state, you cannot get to any valid future state through the interface, which leaves you stateless (i.e., in an implicit state).
    - All explicit states are associated with identifiers. So when in a state, you are acting as the identifier associated with that state, and will be authorized and recorded as associated with and acting as that identifier.
    - How do you get an explicit state? You present an identifier and authenticator to the system, and are given a set of available authorized explicit current states. You can then use those states to produce outputs and provide inputs. One time each. From there forward, you have to get the next explicit state from the output of the previous action. There is no going back, but you can authenticate again...
- An identifier is not like a user identity. There is no explicit association of people to identifiers. Thus there is no implicit identifying information in the system at all. In a similar way to how Kerberos works, you get tickets and you can go on rides. You present an identifier and authenticator, and you get a set of tickets. Every time you go on a ride, you use one of those tickets that is no longer valid for anything ever again.

    - People are often given identifiers and authenticators for short periods, for example to fill out a specific form one time, or over a period of a few weeks. This is used for activities like filling in a document or approving some activity.
    - The identifiers go away, and thus the authenticator is of no use, but the records produced remain in the system of records along with the historical data about the identifiers used to create and update the records.
- The set of tickets (states) you get when you present an identifier and authenticator include tickets for the identifier you presented PLUS sets of tickets for other identifiers that you can "Act As". For example:
    - I can get a ticket that let's me Act As the Security Governance (role) for X (e.g., a company associated with the identifier prefix X). With that ticket (state) I might then be able to update their JDM As-Is SoP information.
    - The system will record the identifier associated with the original authenticator for access as well as the identifier for the security governance role, indicating it was I acting as X.Sec.
    - As a system of records, anything I as X.Sec do is recorded, but all the old versions of any changes are kept and associated with their J as X.Sec. I can change things, but the old versions remain. And old versions can be restored (in Expert mode) so even if I maliciously change things, the previous versions are still there, and all that really happens is that I make a record of the malicious act I performed, and it is readily undone, for example, by J.
- Eventually, I may have to add a system to control the ActAs mechanisms and delegate them to identifiers associated with administrative roles in identifier prefixes (e.g., X.whatever associated with a state of the mechanism allowing any identifier acting as X.whatever to control ActAs for other identifiers with the prefix 'X.'). But that's still some day in the future. Note (FYI) that multiple prefixes (e.g., X.Y.Z.accounting) support a hierarchy of prefixes so that an administrative role for X.Y can potentially add administrative roles for 'X.Y.Z.' prefixes, and possibly also for 'X.' prefixes (and 'I.' Prefixes) depending on their ability to ActAs appropriate other identifiers.

In the meanwhile, as I made these changes, I identified issues in our cash flow simulation system used for business modeling, and naturally decided to fix them for real rather than doing a minor patch to get by…

I warned you… this was more than you ever wanted to know…