Blah Blah,

Four not-so-easy steps to implementing a data-driven security program.

Becoming Data Driven

We wanted to wrap up the book with a handful of things you should focus on as you head back to work. Becoming data-driven doesn’t just mean firing up R or python and tossing in data. Becoming data-driven is an evolutionary process that will slowly shift how you view the world. The value won’t be immediate. Instead, the value will develop over time with punctuated flashes of brilliance. The foundations of a data driven program are as follows:

* Ask questions that have answers in the data
* Find and collect relevant data
* Learn through iteration
* Find statistics

The most difficult part of the transformation is getting started because the first two steps present a chicken and egg problem. You want to ask questions that you have data for and yet you only want to gather data that answer your questions. But don’t worry, through iteration you should be able to build up both.

Ask questions that have objective answers

The opening quote in this chapter was from sabermetrician Bill James. You may know him and his work portrayed in “Moneyball” by Michael Lewis. He challenged much of the conventional wisdom within baseball by leveraging data. The quote is worth repeating here: “My job was to find questions about baseball that have objective answers, that’s all that I do, that’s all that I’ve done.” The focus he has is not on simply exploring and describing the data that is available, nor did he focus on creating colorful visualizations from the data. His focus was purely on finding good questions.

We discussed creating a good question in Chapter 1, but we want to reiterate that a good question has two qualities: it can be objectively answered with data and somebody wants to know the answer. While Bill James could have asked about the effect of stealing bases on player sponsorships, nobody (except maybe the players stealing bases) wanted to know that. He focused on relationships with runs scored, or players on base because those are the questions people wanted answered. The same is true in your work. While you can count blocked spam or create maps covered with botnet infections, if it’s not answering a practical question that someone wants answered it might have been a waste of time.

Knowing that someone cares about the answer can also help shape the question and make the analysis easier. Remember back in Chapter 1, we changed the question from asking how much spam was blocked to asking how much time employees spent dealing with unblocked spam. If, for example, you identified that nothing would change if employees spent less than an hour a week on unfiltered spam, the question then becomes “do employee spend more than an hour a week dealing with spam?” With that threshold in mind, you should be able to simplify the analysis. Rather than calculating how much time, you just need to know if it’s over an hour a week or not. Context and purpose of the question can only clarify the work you do.

Find and collect relevant data

As mentioned at the beginning of this section, data collection and asking good questions have a natural interdependency. The questions you ask depends on having data to answer them, yet you don’t want to collect data you’ll never use. Which comes first? Just from being in your environment you should have some concept of available data: proxy and firewall logs, server authentication logs and even data within the company ticketing system would be good candidates to start with. Start there and form a few practical questions that data can answer. As you get the data to answer your questions, you may need to refine your questions and then you learn more about the data and refine again.

Be prepared to work with others on getting data. Chances are very good you won’t be the custodian for all of the data you’ll want. And so this is the part where we mention that having executive sponsorship is important. If you’re a practitioner, seek executive sponsorship. If you’re in executive leadership, make data sharing happen internally. This will have very limited success as a grass-roots effort. You will need to involve others and probably even reach out across corporate silos in order to get data. You will undoubtedly encounter several objections in some combination of real and imaginary. Keep your eye on the goal though, the effort will pay off in the long run.

type="general"

Information sharing takes a lot more than information

There is a subtle theme across the information security industry that we should all be sharing data and we couldn’t agree more. The initial objection (and a big objection you may run into internally) is a lack of trust and/or a concern about the privacy and confidentiality of the data being shared. This is a valid concern and it’s something that you will have to address. But that’s actually the easy part of information sharing. Sharing information often turns out to be a much larger effort than people imagine. There is an eye-opening moment when the person sharing the information realizes that they have underestimated the amount of time and energy it takes to prepare and share data. There may be some fields that do not or should not be shared and those must be removed. Then there is a validation step to ensure they are only sharing what they intend to share. Finally, storage and transfer of the data may present a challenge in logistics, as the data may be too large to simply email or even to set up a download. The best course is to be open about this and communicate the reality of this to potential partners. The silver lining to work is that the amount of learning laying dormant in the data often more than makes up for the effort of information.

Learn through iteration

Building a data-driven security program may not follow a typical waterfall project plan where the tasks are defined up front and executed one after another. It will be a much more iterative process where each source of data offers its own challenges and opportunities. Iteration becomes the name of the game where setbacks and challenges become just a much a part of the project as success. But do not get discouraged, the setbacks will occur less and less frequently as each one is also a learning opportunity.

One of the big lessons you will undoubtedly learn early on is the importance of data quality and the benefit of building in repeatability. It won’t take long before you pull a data extract and realize a date variable was corrupted, a field was clipped or some other act of nature that requires the whole process be repeated. So not only will the extract, transformation and loading tools need to be automated, data validation processes should be introduced often. You’ll want to realize the integrity of the data was compromised long before you’re generating the final report.

Finally, with the iteration and constant discovery that comes from working with data, you will be forced to check your ego at the door. There is very little room for estimations and guesswork and if things go well, you’ll have this lesson forced upon you over and over. Once the data has proven you wrong a few times, you’ll realize that the data works without motive or agenda and may produce unpopular results. Assumptions should be replaced by questions and data analysis and when things start to come together, you’ll be impress how well they come together and the types of questions you will answer.

Find Statistics

We debated on putting this at the top of the list. Proceeding down a data driven path may head right into the danger zone we talked about in the previous section without some element of statistics involved. The entire point of moving to a data driven security program is to learn from data and the wide field of statistics (encompassing classic, data mining and machine learning) has already learned a great deal about how to learn from data. To not take advantage of all that history may doom you to repeat the failures others have already overcome.

There are two options here: Hire someone with a background in statistics or start enhancing current employees (or yourself!) with training and education. Unfortunately there aren’t many candidates with both good domain expertise and good statistics experience. So hiring external may mean bringing in someone with less experience with information security, which is fine if you are prepared for it. On the other hand, picking up statistics as a working professional will not be possible through some simple weeklong training session. If you take this route, keep in mind the two cultures Leo Breiman wrote about. Some Universities focus on the classic statistics with little focus on programming and data management, while others may focus heavily on programming at the expense of strong foundation of classic statistics.

AR: Hey bob, was thinking of adding in something about “Visualize Everything, describe all your data” but didn’t. This will have to due for now.