Chapter 12: Moving Towards Data Driven Security

“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.”

-- Bill James, Sabermetrician

If you’ve followed along up to this point, you’ve have covered a lot of ground and you’ve hopefully realized that there is knowledge buried in the data. As you begin to move your security practice into a data-driven mindset, we suggest that you take a “panning for gold” approach instead of a “drilling for oil” stance. Meaning that you shouldn’t get bogged down with a single focus (or a single source of data) out of the gate. Instead, roll your pants up, step into the stream of data and just explore and learn what you can about it. Once you understand what’s in the data you can start to ask (and answer) the interesting questions that will begin to make a difference.

This last chapter is dedicated towards that difference. The first half is about moving yourself (or those you work with) towards a data-driven approach at a personal level. The second half is about moving your organization towards a data-driven security program.

Moving yourself towards data driven security

Figure 12-1 is a slight modification of Drew Conway’s “Data Science Venn Diagram” (<http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>), a simple visualization which can help you perform a quick self-evaluation of where you currently are on your journey towards data driven security. We’ll take a look at each major component along with the interactions between some components to give you an idea of where top topics in this book fit and provide pointers for delving into or shoring up areas that may not currently be strengths for you. You don’t have to be strong in the three major areas we discuss here, but you want to be sure the lack of strength in any area is not a weakness that will silently pull you off course.

Figure 12- [f0#.eps]

The Hacker

The term “hacker” has a great deal of confusion surrounding it as it has been usurped by news media and manipulated by marketing firms. In the context of a security data scientist, “hacker” means:

* Being able to code, either scripting in a language like Python or full on programming in something like C;
* Knowing a wide variety of data formats and understanding how to slice, dice, and bend them to your will;
* Having the ability to think logically/scientifically (essentially, not jumping to conclusions) as well as algorithmically (break apart a problem into its composite parts)

If you are an information security professional who isn’t a coder, Chapters 2, 3, and 4 have been designed to help you bootstrap into that skill. If you are a coder, those same chapters cover a language that is most likely new to you (R) and place coding in the context of data analysis versus application building or systems administration, which may be more familiar problem domains for you. Whether you’re at the top of your game as a programmer or just getting started there is always more to learn and no shortage of resources available to do so, including:

**Codeacademy** (<http://www.codecademy.com/>). This is an especially good resource for those new to programming in general or those unfamiliar with a particular language. It would be worthwhile to take a look at the JavaScript & jQuery offerings given the emphasis on JSON in Chapter 8 and D3.js in Chapter 11, and if you don’t know Python well (or at all), their Python course can definitely help.

**Code School** (<https://www.codeschool.com/courses>). The offerings at Code School can be a bit overwhelming and not all are free. However, their R course is freely available at the time of writing and will help you navigate the syntax and nuances of the language.

**W3Schools** (<http://www.w3schools.com/>). If you haven’t had the opportunity to shore up your HTML/CSS/JavaScript skills, W3Schools provides an extremely friendly environment to both learn and experiment. You’ll need at least a basic understanding of these client side components if you want your analyses and results to reach the widest audience.

**StackExchange** (<http://stackexchange.com/>). While you won’t necessarily learn how to code at the StackExchange family of websites, you will have a place to look for answers or ask questions when you’re stumped. Whether it’s trying to understand some esoteric option in ggplot2 or how to do something a bit more complex with a pandas data frame there’s a very good change the answer will be in StackExchange.

When it comes to data formats, security professionals are in the unenviable position of having to be able to manipulate everything from NetFlow captures, to full packet capture (PCAP) dumps and almost every log format known to humankind. The IronPort log file snippet in the MongoDB section of Chapter 8 is an example of how “imperfect” your data world is. While that log file contains highly useful data, it’s in a format that you must parse and convert to make it useful. The only way to get good at that is to actually do it over, and over, and over again, building up reusable bits of code and techniques along the way to save you some time later on.

Learning how to think logically, scientifically and algorithmically requires time, effort and practice. Formal, in-person, instructor-led education may work best for some students, especially those who have shied away from programming. However, introductory sites like Project Euler (<http://projecteuler.net/problems>) can get you started down this path, more advanced and diverse problem sets can be found at Kaggle (<http://www.kaggle.com/competitions>), and you can delve into wide and deep security domain problems at the VAST Challenge (<http://vacommunity.org/VAST+Challenge+2013>) site (look in both current and previous years’ sections).

Overarching these three traits is the need to develop and hone a sense of *curiosity*. In fact, curiosity may be the **single most important trait** of a “hacker”. The need to know *why* or *how* something works the way it does from start to finish is an invaluable driving force when faced with a complex data science problem. When combined with the other two security data science primary skills (statistics knowledge and security domain expertise), you can eventually get to a place where developing a successful NetFlow-based malware traffic clustering algorithm is as rewarding as pwning the other team in a capture-the-flag competition.

type="general"

Developing Developer Skills

While we offer resources in this section to help you pick up the skills necessary to write code, but there are skills *around* writing code that come in handy as a code warrior. Two of the most not-so-secret skills you should also develop are with unit tests and source code control. Becoming comfortable with writing and executive unit tests tighten up not just your code, but how you think about your code. While yes, you are a brilliant person with amazing skills, you will make mistakes and logic errors in your code despite that fact. Implementing unit tests will help you catch those inevitable oversights that will creep into your code. Along the same lines, source code control helps track and when multiple developers are working on the same code but also enable more advanced features such as version control and branching of the code. But more than that, source code repositories also help avoid that awful question of “now where did we put that source code?”

The Statistician

Given some of the “rookie mistakes” seen in many security industry reports and the prevalence of raw counts in security dashboards there’s a high probability that statistics may be the weakest area for information security professionals. We’ve covered some statistical concepts in depth and provided a whirlwind overview of others in Chapters 4, 5, 7, and 9. A PhD in statistics is not required to be an effective security data scientist, but it’s important to have an understanding of the fundamentals of statistical analysis and machine learning, even if you’re part of a multidisciplinary team.

While you can head over to your local college or university and dive into a traditional classroom program there are two other options to consider when looking to obtain a better understanding of statistics:

Massively Open Online Courses (MOOCs) like Coursera’s Data Science course (<https://www.coursera.org/course/datasci>), edX’s Learning From Data course (<https://www.edx.org/course/caltechx/cs1156x/learning-data/1120>) and Syracuse University’s Data Science Open Online course (<http://ischool.syr.edu/future/cas/introtodatasciencemooc.aspx>) provide a low-risk way to plug into a formal statistics curriculum but aren’t right for everyone. Lectures, handouts and assignments are available at your convenience (within a course’s overall schedule) and discussion forums provide a way to interact with professors, teaching assistants, and fellow students, it can be bit overwhelming or even distracting to be in a setting with 2,000 to 4,000 individuals. Individual attention can also be difficult to obtain if you’re struggling. Employers and professional organizations may also not yet accept the certifications from MOOCs, making the time investment more for personal benefit than professional credential gains.

Online certificate or Masters courses such as UC Berkeley’s MIDS (<http://www.ischool.berkeley.edu/programs/mids>) program, University of Washington’s certificate in data science (http://www.pce.uw.edu/certificates/data-science.html) and Penn State’s Applied Statistics online curriculum (<http://www.worldcampus.psu.edu/degrees-and-certificates/applied-statistics-masters/course-list>) offer the structure and size of a traditional classroom with the convenience being online.

Understanding and applying statistics correctly is more complex than you might imagine, and individuals in disciplines with a rich history of using statistics to solve complex problems often times fall into common traps. Resources such as Alex Reinhart’s *Statistics Done Wrong* (<http://www.refsmmat.com/statistics/>) and DZone’s mis-named “Big Data” Machine Learning reference (http://refcardz.dzone.com/refcardz/machine-learning-predictive) are good to keep on hand to keep your analyses on-track.

The Security Domain Expert

When focusing on the topic of security domain expertise as it relates to data science, “thought leaders”, “gurus”, and “rock stars” need not apply. What we’re really talking about here are practitioners with solid, in-the-trenches, real-world experience. Depending on your area of focus (information security covers a broad range of topics), you may be applying your combined hacking skills, statistics knowledge and expertise to:

* Develop smarter endpoint protection system algorithms
* Discover new ways to detect anomalous behavior in network data
* Uncover patterns from vulnerability assessments to help determine why some systems fall out of compliance more than others
* Provide meaningful, and useful metrics for various components of your overall security program

or a host of other areas.

**Your** insight is, perhaps, **the most valuable component** to this data science triad, as it will move computations sans context into the realm of analyses driving action. There is virtually no way for an organization or individual to effectively crunch “security data” without this domain expertise. Your assistance and knowledge is vital in crafting clever questions and confirm results. Your insight into the networks and systems of your organization, behaviors and characteristics of malware, and classification and qualification incidents will be the critical factor in corresponding analyses.

The Danger Zone

A little knowledge is a dangerous thing, and having a basic ability to gather and programmatically crunch data backed by a bit of industry knowledge is where you may fall into the trap of thinking your doing data science when all you’re doing is reputational damage to all three component areas (and, potential, yourself). How do you steer clear of the danger zone?

**Embrace (versus dabble in) statistics**. Statistics and machine learning has enabled advancements in everything from a deeper understanding of the microscopic workings of our genes, to telling us how many steps and flights of stairs we’ve taken to how to build spacecraft that eventually break pas the limits of our solar system. They can absolutely help enhance our knowledge of security issues and even help solve some of them. Just don’t think you can dip your toe in. Not everyone can be a PhD in statistics, so make sure to surround yourself with a physical or virtual team with at least one strong stats person to help you stay on course.

**Dig deep but stay wide**. You will need to know certain aspects of information security just as thoroughly as individual biologists know the deep vertical segments of their discipline. But, because so many areas outside “security” (e.g. economics, politics, human rights) have an impact on “security” you’ll need to factor those in as you move from asking “What?” and “How?” to “Why?” and “Who?”. Finally, there’s a reason the CISSP certification has ten domains. You can’t be an expert in each, but you should know enough about each of them to bring in expert help when needed.

**Challenge assumption and validate results.** Hold yourself and ask others to hold you accountable all the way through your analyses. Whether you’re working on internal organizational data or performing research you intend to publish and/or speak about pair up with practitioners who can help you keep on the straight and narrow path. When you’ve released your findings, take an example from the reproducible research (http://www.foastat.org/resources.html) movement and ensure there is sufficient documentation and data available for others to test out your findings.

Moving your organization towards data driven security

By now you realize that 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 and those in your organization view the world. The value will not be immediate. Instead, the value will develop over time with punctuated flashes of brilliance. The components of a good data driven program within any organization have some combination of the following:

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

The most difficult part of the transformation is getting started because the first two components 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 we mentioned in the first part of this chapter. Unfortunately the candidates with both good domain expertise and good statistics experience are few and far between. 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 a simple weeklong training session. If you are seeking educational programs in statistics, keep in mind the two cultures Leo Breiman wrote about. Some Universities focus on the classic statistics with less (or no) focus on programming and data management, while others may focus heavily on programming at the expense of teaching a strong foundation of classic statistics.

Summary

We have covered a lot through the pages of this book and you should realize that you don’t have to do all of this right out of the gate. Through the mixture of hacking skills, domain expertise and statistics you can move yourself towards a data driven lifestyle. Combining that with the art of asking the right questions and getting the data to answer those questions will start to move your organization towards a data driven security program. You don’t have to implement everything right away to see value. An iterative approach should provide more value over time and help you adapt to the inevitable challenges that will arise. Start slow, try everything, try everything again and let us know how you’re doing.