[[copy edited by Kezia Endsley]]

Chapter 12: Moving Toward Data-Driven Security

[AU: Lots of good general principles in this chapter, and it reads very well. However, holistically, I wonder if you have any specific experiences you could relate to illustrate some of these points (maybe as sidebars). Any specific stories you could offer to illustrate these principles in action would be good additions here. Thanks, Kevin (PjE)]

[AR: acknowledged, will attempt to expand a bit. ]

“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 been following along up to this point, you 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 to that difference. The first half is about moving yourself (or those you work with) toward a data-driven approach at a personal level. The second half is about moving your organization toward a data-driven security program.

Moving Yourself Toward 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>), which is a simple visualization that can help you quickly evaluate where you currently are on your journey toward data-driven security. This chapter looks at each major component, along with the interactions between some components. The idea is to help you identify areas that aren’t currently your strengths. You don’t have to be strong in all the major areas discussed here, but you want to be sure that weakness in any one area doesn’t silently pull you off course.

Figure 12-1: The Data science venn diagram [793725 c12f001.eps]

Licensed under Creative Commons Attribution-NonCommercial (<http://creativecommons.org/licenses/by-nc/3.0/legalcode>)

The Hacker

[[Authors: I can see that by modifying Conway’s diagram, you are attempting to appeal to readers who view themselves as “hackers” rather than as “programmers” or “computer scientists”. But I think “hacker” is not appropriate the way you are using it. “Hacker” connotes the ability to “do the unexpected”, including finding vulnerabilities or hidden functionality or ways to break systems. Given your three bullet points below, what you have described a “programmer” or “coder”. That would be a much better term to use – both to avoid confusion and to communicate your points better. For example, in the Conway diagram, “data science” is not at the intersection of “statistics” and “hacking”. But it is appropriate to say that “data science” is at the intersection of “programming” and “statistics”. TE]]

[AR: we talked with Russ, he thought we changed the original diagram, but we will try to emphasize the “hacker” side a bit more]

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, we are updating the classic use of “hacker” to describe someone with a passion for using (and perhaps abusing) technology for a benevolent goal, including skills such as:

* Being able to command computers through code, either via 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 critically, logically, scientifically (essentially, not jumping to conclusions) as well as algorithmically (break apart a problem into its composite parts)
* Being able to communicate your work through visualizations, charts, tables or even a good old-fashioned collection of words.

The Coder

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. There are no shortage of resources available, including:

* **Codecademy** (<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’s worthwhile to take a look at the JavaScript and jQuery offerings given the emphasis on JSON in Chapter 8 and D3.js in Chapter 11. 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 this 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 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/>)—Although 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 doing something a bit more complex with a pandas data frame, there’s a very good chance the answer will be in StackExchange.

The Data Munger

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. Although that log file contains highly useful data, it’s in a format that you must parse and convert to make useful. The only way to get good at that is to do it over and over again, building up reusable bits of code and techniques along the way to save time later.

We also would be remiss if we didn’t include a warning in this section. We can tell you from experience, while data analysis is absolutely about the analysis, that’s not where you will spend most of your time. Most of your time will be spent transforming, cleaning and preparing data. That task is at its core, a combination of the previous section on coding and the next section of thinking, as both of those skills will be used to extract the useful data and prepare it for analysis. Though the skills of security domain expertise and statistics will also help to identify what you want to keep and how you want to clean it to be helpful in the analysis. For a list of helpful tools for this task, see Appendix A and the section on “Data Cleansing.”

The Thinker

Learning how to think critically, 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 the current and previous years’ sections).

These resources will supply data in various states. One of the criticisms of competitions like Kaggle though, is that they offer the data in a ready-to-be-analyzed format. As we mentioned in the last section, this is very much unlike the real world and so just focusing on things like the Kaggle competitions may give you skewed perspective of the real world. As a stark contrast, the VAST Challenge has constructed real-world logs and device outputs that must be cleaned and prepared prior to analysis, thus giving a better idea of what real world data is like.

The Visualizer

The skill to communicate to outsiders was never part of the original use of the term “hacker” though it certainly is evolving into that. It’s not enough to make the technology bend to your will and make a discovery. You must also be able to communicate that in a language that the audience can not only understand, but also relate to and appreciate. While this skill may mostly be about data visualizations skill (and all of Chapter 6), that shouldn’t be the only tool in your toolbox. Realizing when to scrap those glossy color pictures and produce a simple table or even just describe the results in an email or in person is more valuable sometimes than the data visualization skill itself. Many resources are freely available and doing a quick search over the internet will lead you to far more resources than we can list here. However here are a few to get you started:

* **Flowing Data** (http://flowingdata.com) Not only does the maintainer Nathan Yau provide some incredibly inspirational data visualizations, he will also include a few comments and insights into the data. Having written two books on the subject of data visualization Yau knows good data visualizations!
* **Junk Charts** (http://junkcharts.typepad.com) Because sometimes knowing how *not* to create a visualization is more helpful than knowing how to create one.
* **Storytelling with Data** (<http://www.storytellingwithdata.com>) One of the great things about this site is that Cole Nussbaumer has a very pragmatic approach to visualizations and will talk about visualization makeover and the processes used so almost anyone can follow along and learn.
* And there are other sites that aggregate visualizations and are good to keep an eye on such as http://visualizing.org, http://visual.ly, or http://eagereyes.org and there are plenty of other tools and resources in Appendex A.

Overarching these 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 main security data science skills (statistics knowledge and security domain expertise), you’ll eventually get to a place where developing a successful NetFlow-based malware traffic clustering algorithm is as rewarding as beating the other team in a capture-the-flag competition.

[[Authors: I can see value in this paragraph of tying the trait “curiosity” to the “hacker” label. There is a depth of diagnostic/investigative skill that hackers develop that “mere” programmers don’t always have.

Therefore, given my comments above, you might adopt the term “Programmer/Hacker” rather than just “hacker” alone.

[AR: I think it’s nice to just mention this as a wrap up to this section.]

type="general"

Developing Developer Skills

Although the resources in this section can help you pick up the skills necessary to write code, there are skills *around* writing code that come in handy as a code warrior. Two of the not-so-secret skills you should develop are unit testing and source code control.

* Becoming comfortable with writing and executing unit tests tightens up not just your code, but how you think about your code. Yes⎯you are a brilliant person with amazing skills⎯but you will still make mistakes and logic errors in your code despite that fact. Unit testing helps you catch those inevitable oversights that creep into your code.
* Along the same lines, source code control helps track multiple developers’ code efforts, and enables more advanced features such as version control and code branching. More than that, source code repositories also help you avoid that awful question, “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 is the weakest area for information security professionals. You learned about some statistical concepts in depth and read a whirlwind overview of others in Chapters 4, 5, 7, and 9. Okay, you don’t need a PhD in statistics to be an effective security data scientist. However, it’s important to have an understanding of the fundamentals of statistical analysis and machine learning, even when you’re part of a multidisciplinary team.

Although 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 you want a better understanding of statistics:

* Massively Open Online Courses (MOOCs) like Coursera’s Introduction to 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 master’s courses such as UC Berkeley’s MIDS program (<http://www.ischool.berkeley.edu/programs/mids>), 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-certificate/overview) offer the structure and size of a traditional classroom with the convenience being online.

[[Author: Can you please fill in the missing URL above? Kezia]] //Yes, authors, the Univ. of Washington one is missing. Thanks, Kevin (PjE)

[AU: The Penn State URL doesn’t bring up anything for me. Please check and fix. Thanks, Kevin (PjE)

[AR: added link and fixed Penn State]

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 oftentimes fall into common traps. Resources such as Alex Reinhart’s Statistics Done Wrong (<http://www.refsmmat.com/statistics/>) and DZone’s misnamed “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 I’m 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 confirming results. Your insight into the networks and systems of your organization, the behaviors and characteristics of malware, and the classification and qualification incidents is the critical factor in corresponding analyses.

The Danger Zone

A little knowledge is a dangerous thing, and having the basic ability to gather and programmatically crunch data, along with a bit of industry knowledge is tricky. Don’t fall into the trap of thinking you’re doing data science when all you’re doing is reputational damage to all three component areas (and, potentially, yourself). How do you steer clear of the danger zone? Try these approaches:

* **Embrace (versus dabble in) statistics.** Statistics and machine learning have enabled advancements in everything from a deeper understanding of the microscopic workings of human genes, to telling you how many steps and flights of stairs you’ve taken, to building spacecraft that eventually break past the limits of the solar system. They can absolutely help enhance your knowledge of security issues and even help solve some of them. Just don’t think you can dip your toe in. Not everyone can become a genius with statistics, but make sure your team (physical or virtual) has at least one strong stats person.
* **Dig deep, but stay wide.** You 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 (for example, economics, politics, and 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 10 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 assumptions and validate results.** Keep an open mind, because data has a way of changing your mind for you. 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 movement (http://www.foastat.org/resources.html) and ensure there is sufficient documentation and data available for others to test your findings.

Moving Your Organization Toward Data-Driven Security

By now you realize that becoming data-driven doesn’t just mean firing up R or Python and tossing in the 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 (again)

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, yet you only want to gather data that answer your questions. But don’t worry; through iteration, you can 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 the book *Moneyball* by Michael Lewis. He challenged much of the conventional wisdom within baseball by leveraging data. Recall that he said, “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.” His focus was 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 finding good questions that have answers in the data.

Chapter 1 discussed creating a good question. Remember that a good question has two qualities⎯it can be objectively answered with data and somebody wants to know the answer. Although 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. Although 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 be 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 employees 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. 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 depend 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 are all good candidates to start. 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 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 of all of the data you’ll want. This is why 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 grassroots effort. You 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 push across the information security industry that we should all be sharing data, which is a good thing. 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, even internally within an organization. This is a valid concern and it’s something that you 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 people sharing the information realize 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 sharing only 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 for downloading. The best course is to be open about these challenges and communicate the reality to potential partners. The silver lining is that the amount of learning you can do when you share data often more than makes up for the effort to share it.

Learn Through Iteration

When you’re building a data-driven security program, you won’t follow a typical waterfall project plan where the tasks are defined up front and executed one after another. It’s a much more iterative process where each source of data offers its own challenges and opportunities. Iteration becomes the name of the game and setbacks and challenges become just as much a part of the project as success. But do not get discouraged; the setbacks will occur less and less frequently and each one is 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 requires that 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 that 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. 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. When things start to come together, you’ll be impressed about the types of questions you can answer.

type="general"

Building A Real Life Security Data Science Team

There is an allure surrounding data science. Those who wield its magical mathematical and technological incantations are treated with a reverence akin to the gurus of early days of UNIX. Yet, this mystique quickly wears off when you move from thinking about building a data science team to actually acting on that impulse.

When Bob had his internal team start their move into security data science it was difficult to resist the urge to spin up a giant, shiny Hadoop cluster and start importing every log from every system into a massive data store. In truth, his team did start down the Hadoop path and found it fraught with peril (and screenfulls warning messages).

Rather than focus on the technology, they stopped and focused on defining what single question (to improve security capability) they would like answered if they had the data. Not five. Not three. One. That single question ended up being a statement: “We want to be able to search for an indicator of compromise (in this case, an IP address or IP/port combination) across **all** our perimeter devices in less than five minutes.” For most organizations—including Bob’s—the total volume of such data would fit well within the category of “medium-sized” (i.e. not “big”) data.

His team focused on using a traditional SQL (MariaDB), NoSQL (MongoDB & Redis), R, Python and JavaScript. Bob had one of his staff attain the University of Washington’s Data Science certificate to ensure there was a healthy mix of formal analytical knowledge and domain expertise on the team. For six long months, they iterated from failure to failure, trying different ways to acquire, clean and store data (they call that *data curating*), and structure schemas and formulate queries. Along the way, they suffered setbacks when log file formats changed without warning, when data access issues cropped up, and when the absolute need for referential metadata reared its ugly head.

Three core principles focused the team. First, master the open source versions of tools before engaging vendors. If you don’t know how the sausage is being made, you really have no idea what’s being done and this is vital when working with real data. Second, follow the mantra of “no single tool; no single database; and, no single approach to solving a problem”. Do not put blinders on because you are either comfortable with certain technologies or have an affinity for a certain tool. Third, failure is expected, but you must learn from each journey down the wrong path, much like a mortar team must adjust angle and deflection after each errant shot.

Ultimately, Bob’s team met the five-minute challenge and has moved on to other questions. Your team—and, just like firing a mortar, it is a team effort—will also be successful if start with a question, be iterative and methodical in your approaches and never stop learning, both from your mistakes and the successes of others.

[AU: Another illustration would be a good addition to this chapter. Do you have a visualization of the iterative process you are talking about here that you could share? One that might show readers what this process is like? Thanks, Kevin (PjE)]

AR: Done. Thx.

Find Statistics

We debated on putting this at the top of the list, but hopefully we’ve pounded this point home by now. 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. The wide field of statistics (encompassing classic, data mining, and machine learning) has already learned how to learn from data. Not taking 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, candidates with both good domain expertise and good statistics experience are few and far between. So hiring externally may mean bringing in someone with less experience with information security, which is fine if you are prepared for it. On the other hand, becoming a professional statistician isn’t 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, whereas others focus heavily on programming at the expense of teaching a strong foundation in classical statistics.

Summary

You have learned 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 toward a data-driven lifestyle. Combine that with the art of asking the right questions and getting the data to answer those questions, and you’ll start to move your organization toward 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 arise. Start slow, try everything, try everything again, and let us know how you’re doing.

**Recommended Reading**

[AU: For consistency with other chapters it would be good to add a couple recommended readings to this chapter. Thanks, Kevin (PJE)]