CHAPTER 1

Introduction: The Manifesto and the BOOM! Framework

I don't believe in astrology; I'm a Sagittarius and we're skeptical.

— Arthur C. Clarke

We are scientists. We don't blog. We don't twitter. We take our time… bear with us while we think. Comical words found in *The Slow Science Manifesto*[1] written by none other than The Slow Science Academy. Their manifesto serves first and foremost as a reminder to themselves. It defines who they are and aspire to be. It also serves to set public expectations. In this case, "don't expect much, and certainly not via Twitter." And yet they humbly advocate for a point of view, "*Science needs time to think. Science needs time to read, and time to fail. Science does not always know what it might be at right now.*" A good manifesto does all these things. It creates identity for its signees, it sets expectations for its audience, and it advocates for a point of view without being too much of a bully.

The Metrics Manifesto strives to do all those things, particularly the last one – a point of view without too much tyranny. It endeavors to be a framework for creating simple security metrics and advanced ones for those who need them. Lastly, this book serves as a guide for making a complete enterprise security metrics program – a program that is grounded in the principle of confronting security with data.

One caveat before you start: A manifesto typically outlines a minority position. You wouldn't need a manifesto if everyone already agreed with you. And while this is a minority position relative to security, it's likely a majority position with measurement professionals at large. The term *measurement professionals* includes scientists, actuaries, statisticians, engineers, and others. This is the group of people who seemed to align with our previous book titled *How to Measure Anything in Cybersecurity Risk*

(Wiley 2016). The actuaries really liked it! In 2018, it was required reading for The Society of Actuaries exam prep. While the Manifesto is quite different from that work, it fully aligns in measurement spirit. Be forewarned: The methods herein may be foreign and at times challenging. When you feel unsure, just imagine measurement experts past and present cheering you on!

Lastly, I do hope "The Manifesto" produces *productive skepticism* about security. It should come naturally to us. After all, security professionals poke and prod to discover why someone else's digital ideas are risky. Shouldn't a true skeptic turn that same confrontational mindset on themselves and muse, "*This security capability I've deployed may not work. What would I see occurring that would let me know if it does?*" That's the first step in confronting security. It's the first step in designing a powerful security metrics system that makes a significant difference in our battle against our adversaries.

What's Next: Caveats and Epiphanies

The next section covers the manifesto and the BOOM Framework. The manifesto is built around four key observations. Each observation, in turn, has one or more supporting beliefs. You don't need to become a convert to those beliefs. In fact, you should maintain doubt. That would line up with the theme of "confronting security with data."

The BOOM Framework is built around five key baselines. And these five baselines each get hefty chapters dedicated to them. Now for some caveats.

The first caveat is that skipping chapters will be rough without the right background. That's why I recommend reading the whole book.

My next caveat is the same one Doug Hubbard and I made in our last book: *This is not a statistics book.* What is it then? It's a metrics book. I know it seems obvious to say that – but I think certain readers appreciate being forewarned. If you are coming here looking for the latest, greatest, in-depth quantitative stuff, then this may not be the book for you. That being said, there may be some perspectives that even seasoned data scientists, statisticians, and others may find of interest.

Next to the last caveat: This book has code. The good news is that much of the code is in the form of one-liners (some lines might be quite long due to clever tricks of the trade). It's not my plan to turn you into a data scientist. First, I don't think I am qualified. Second, it's completely unnecessary. Why become a carpenter when you only need to use a hammer?

Last caveat: This one is on my qualifications to write a book with such a lofty title. My qualifications are that I am well acquainted with operational sadness. I've spent most of my career in the foxhole – both on the vendor side and in operations at varying organizational levels. The whole time, I couldn't shake the feeling that there must be a better way to manage operations – particularly

security. Dissatisfied, and prone to wander, I started to look outside of security and even technology. My question was, "*Who else was solving big problems where uncertainty abounds and the risks are real?*" This is when I started running across people I will refer to as *measurement experts*.

Measurement experts are the humble statisticians, natural scientists, decision analysts, and other folks tackling seemingly impossible-to-measure problems. They are all decidedly more educated than me, and a few can sling code really well. But what I bring to the kitchen table, and you do too, is operational experience within a problem domain. Once I relaxed my prejudices about my lack of quantitative savvy, I started to become productive – dare I say creative. This newfound freedom led to the following "epiphanies of the obvious."

Epiphanies of the Obvious

- **Computers are very good at math.** So good at it, in fact, that you don't need to be. Understanding calculus has great utility, but your computers often know enough for most of your problems.
- **Measurement experts create their wares for other people to use.** This is similar to the cobbler who makes shoes for others to wear – just not for himself. There is a vast array of freely available analytic tools made for people like you and me.
- **Problem understanding is job #1, not quantitative skills.** My favorite quote by Charles Kettering reads, "*A problem well defined is a problem half solved*." Once you have framed your problem well, you can usually pull down the code you need from the spice rack of analytics to start baking. If you happen to have a problem that is out of reach, call an expert – 99% of the problems you have are likely in reach if you spend enough time framing your problem and being resourceful.
- **The last epiphany (which we covered in the first book) was that our goal should be to "beat the competing model."** I just need to be slightly better than the competition, not perfect. In short, when you think you must have perfect math, code, statistics, i.e. "metrics" then you are bound to the fate of Sisyphus. He was the king cursed to push a boulder up a hill for all eternity. I think this mindset may be the blocker that prevents security teams from taking things to the next level in operational excellence, I am very sad to say.

Thus, my ambitions for this book, and for you, remain humble. I merely want to beat the competing model for security metrics. That model is typically just a list of basic counts of things and not much more. Don't get me wrong – lists of metrics and counts are not necessarily bad. In fact, they are necessary. I just know we can do better together.

Next up is "The Manifesto" and the BOOM! Metrics Framework.

The Metrics Manifesto and BOOM!

A lot of people have problems with public confrontation, but it doesn't worry me at all. I can handle myself. I know my martial arts. – Pink

This section presents the pithy "Metrics Manifesto." It's less than a page in length. Think of it as the ethos, or spirit, behind the book. I encourage re-reading it from time to time.

The rest of the chapter outlines the BOOM! Framework. It's the metrics framework that evolved out of the aforementioned speed consulting. It also provides an outline for the book. With BOOM, you will encounter interesting measurement methods like survival analysis, burndown rates, arrival analysis, interarrival analysis, escape rate, Bayesian data analysis, and more. Each method is designed to help you confront your security program with data. Taken as a whole, these methods embody the Metrics Manifesto.

The (Modern) Metrics Manifesto

Observation: Most metrics count; the best ones confront.

Belief: *"We believe shrinking attack surface, while not slowing value exposure, is the new job #1 for security."*
Belief: *"We also believe not doing this gives advantage to our adversaries and reduces business opportunity."*

Observation: Most metrics reveal what is certain; the best ones also retain what is uncertain.

Belief: *"We believe metrics that ignore our uncertainty ignore our adversaries."*

Observation: Most metrics focus on beating benchmarks; the best ones focus on beating the competing model.

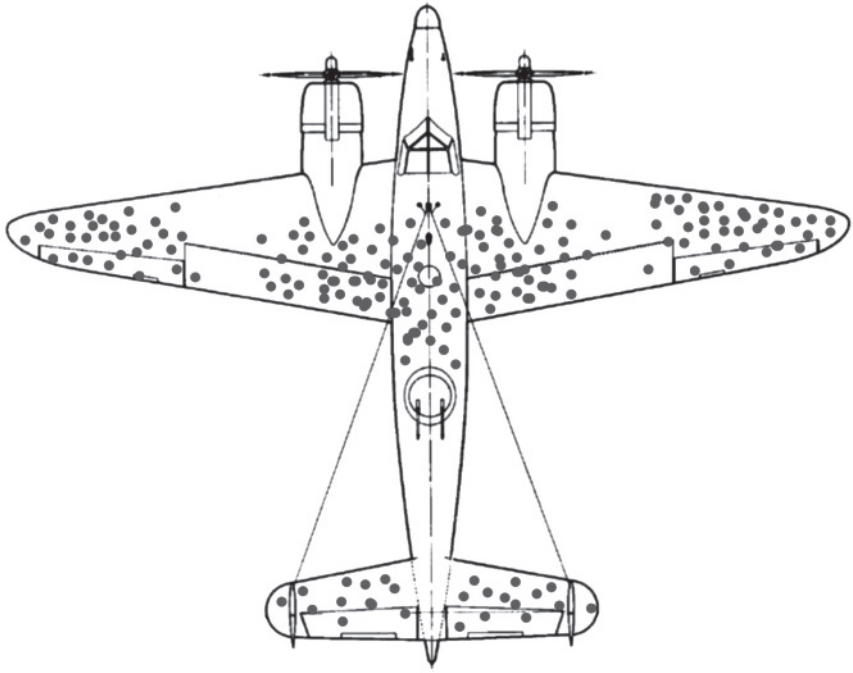
Belief: *"We don't believe in benchmarks (for the most part), and neither do our adversaries."*
Belief: *"We believe in continuous improvement because our adversaries attack, and our business expose value…continuously."*

Observation: Most metrics require data; the best ones can start with little or none.

Belief: *"We believe resourcefulness with small data is always better than complexity with big data."*
Belief: *"We also believe expertise can be turned into data when you have none – data, that is."*

BOOM: Baseline Objectives and Optimization Measurements

The winners are usually the guys who get 5% fewer of their planes shot down, or use 5% less fuel, or get 5% more nutrition into their infantry at 95% of the cost. – How Not to Be Wrong[2]



Bullet Holes and Bombers

During WWII, the US Navy continually lost bombers to anti-aircraft artillery. In hopes of building a more resilient aircraft, officers suggested they examine all planes returning from battle. They came to the conclusion that reinforcements should go above the wings and tailpiece, where there were more bullet holes (see image above).

That approach seemed reasonable to everyone … except for one statistician by the name of Abraham Wald. Wald believed that the secret to building a more anti-aircraft-proof bomber lay in the planes that *didn't* make it back. It turns out Wald was right. Planes that had damage in low-impact areas (as seen in the figure) survived. Planes that didn't make it back had bullet holes in critical areas, where pilots sat and where the engines were located.

What was the problem here? The Navy officers were using the wrong object of measurement. If they had followed through with their plans, armor would have gone everywhere but where it was needed. The engines and cockpit would have stayed exposed, leading to a colossal waste of life and money.

We discuss the problems of having the wrong object of measurement in our first book. It is a frequent and costly error that can plague whole industries. How could battle-hardened military experts be so wrong about something so important?

The Ease of Self-Deception

We confuse what's *easy to measure*, like bullet holes in returned planes, with what is *important to measure*, like bullet holes in downed planes. We deceive ourselves by measuring only what is obvious, forgetting the goal of our measurement. Wald's statistics training prepared him to catch this form of self-deception.

As security people, the question we have to ask is, "*Are we measuring the right things?*" Or, have we been deceived? If we only count security events without considering our objectives, then the answer is likely, "Yes we have been deceived!" The next section introduces BOOM – a framework to help you avoid deception and focus on capability measurement and objectives for betterment.

BOOM Defined

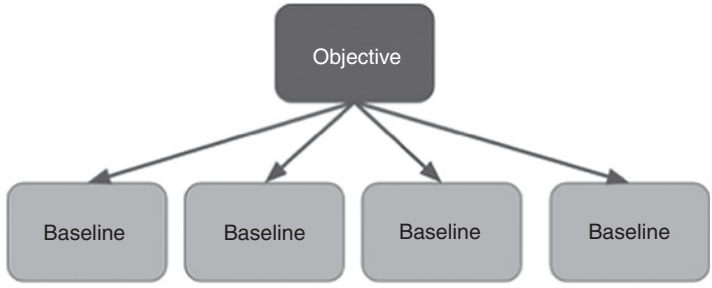
BOOM stands for Baseline Objectives and Optimization Measurements. Its ingredients include:

- Five baseline measurements
- A KPI-based scoring system

- Reusable data structures
- Simple dashboard objects

The first component of **B**OOM is **B**aselines – the fundamental measurements behind the BOOM framework. These metrics baseline your capabilities over time. They reveal if your capabilities are optimizing (improving), scaling (keeping up), or degrading.

The second concept in B**O**OM is **O**bjectives. Objectives define the capability "outcome" and its goal for improvement. An objective is similar to key performance indicators (KPIs). An example objective might be: *Reduce the time to live of customer facing exploitable vulnerabilities by 80% by the end of Q4*. You would use one or more baselines to measure this objective.



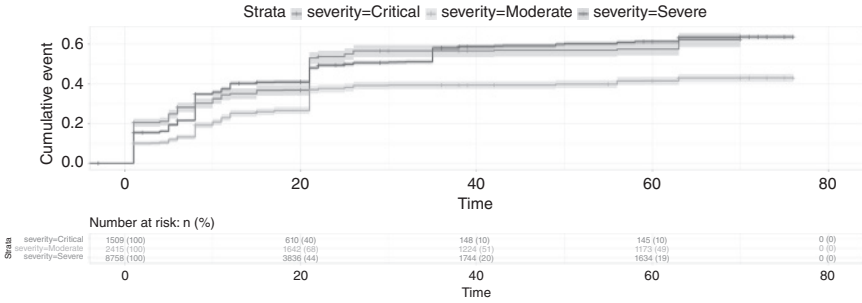
The next section defines each of the five baselines starting with survival analysis.

Survival Analysis

Survival analysis, TTL analysis, and engineering failure analysis are all related analytics for measurement of event survival times. In this case, they measure the complete range of survival times as a curve (seen below). As you will learn in Chapter 2, survival analysis dates back 400-plus years. Today it's used prominently in epidemiology and actuarial sciences.

The goal is a more complete and data-rich answer to how long risk survives. For example: "50% of critical vulnerabilities live for 48 hours or longer, 10% live for two weeks or longer, and 1% live for one year or longer." You may have an improvement in the average TTL of critical vulnerabilities while seeing those at the 1% ranges growing in age.

Plain averages obscure your capabilities' true performance. That's why not measuring this way leads to uncertainty that gives advantage to the bad guys.

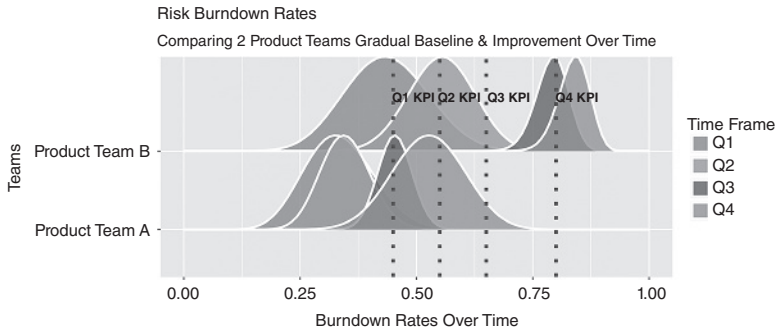


Burndown Metrics

Burndowns are a ratio of risk removed over total risk. Imagine in January you had 100 new critical vulnerabilities added and 50 removed. That is a 50% burndown rate. Next month, there are 10 new and 25 fixed. The overall, or cumulative, burndown for the two months is 68% (75/110) with a positive trend of 18%.

The following graph is an example of BOOM at work. It's one of many visualizations for measuring quarterly KPIs. The vertical dotted lines are KPI targets. The distributions (bell-shaped objects) show our uncertainty.

If the far-right distribution for Team B could talk, it would say, "*The real burndown rate is around 85% but it may be closer to 75% or 90% . . . if you want more certainty I need more data!*" Don't worry if this seems a bit Greek to you. By the end of the book, and particularly Chapter 3, it will seem normal.



Measuring our uncertainty about meeting KPIs reflects the second manifesto observation, "*Most metrics reveal what is certain, the best ones also retain what is uncertain*." When you are measuring capabilities, you are measuring rates over time.

There will always be some uncertainty when it comes to rates. Consider batting averages. Batting averages are based on an accumulation of data over a time frame. In five games, you may bat .400. Over the last three years, you batted .270. Without a broader time frame, you can get fooled due to small data.

In the graph above, Product Team B is largely meeting their Q4 objective. I say "largely" because the distribution is ~20% to the left of the last KPI line. Team B clearly met their Q3 KPI. We know this because there is zero overlap of the Q3 distribution with the third KPI line.

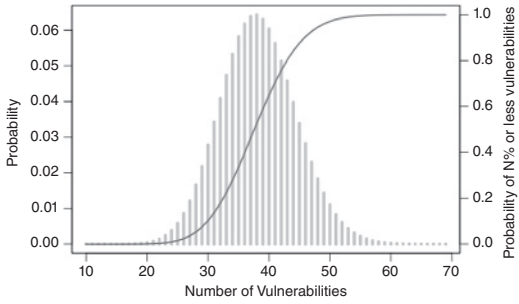
Note how Team B's Q3 and Q4 distributions overlap. The fact is, Q3 and Q4 may be nearly identical, given our uncertainty. Does that mean burndown rates haven't moved much in the last six months? Perhaps. The big gains were between Q2 and Q3.

One last observation. It is entirely possible that Product Team A (row below B) made no real progress the whole year. Look at how much the distributions overlap. You will be introduced to methods to help you credibly gauge the differences in data rates.

Arrival Rates

Burndown rates measure your capabilities for getting rid of risk. It's considered a "right of boom," or shift right, measurement – meaning, burndown measures processes that occur after a bad thing has happened.

Arrival rates are "left of boom" – or "shift left." You are baselining the rate with which risk materializes, with the idea being that you can implement risk-prevention capabilities.

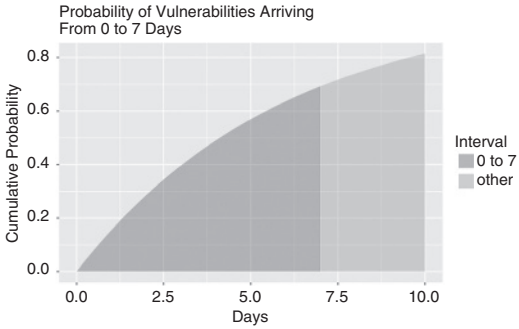


It may seem counterintuitive, but things like vulnerabilities tend to have consistent arrival rates over time. In Chapter 6, which focuses on interarrivals, you will apply "Open Source Intelligence" (OSINT) on the National Vulnerability Database (NVD). Spoiler, what we also learn is that the arrival rates for extreme vulnerabilities do not fluctuate wildly year over year.

Arrivals have an impact on your organization. You have to respond to the rate with which risk materializes – be it third- or first-party sourced.

You will learn how to create and interpret graphs like this one that measure the probability of vulnerability arrivals. Based on the data, there is a 60% chance of 40 extreme vulnerabilities (or less) showing up over the next 365 days. It is retaining our uncertainty without obscuring our certainty.

Wait-Times (Interarrival Time)



Arrivals measure how much risk shows up in a given time frame. A related measure is the time between arrivals, called interarrival times (aka wait-times). This is the focus of Chapter 6.

Interarrival time is a common measure in operations management. It is used for queue measurement and optimization. The "things" in our queue are threats and vulnerabilities. If we are going to effectively manage the queue, we need to measure what's in it.

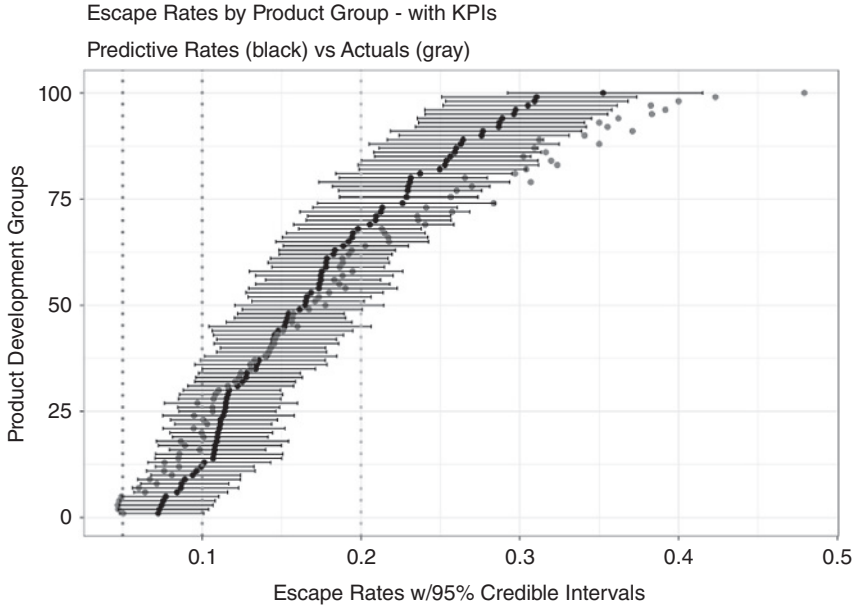
A leading risk indicator would be a decrease in wait-times. This graph shows a predicted wait-time between extreme vulnerabilities. It is forecasting the probability of one event in seven days or less. It's roughly 70%. This could also be applied to any type of threat – phishing, ransomware, or any myriad of event types you are looking to control. In a sense, you have queues for each of these risks.

Escape Rates

There is a rate with which risk moves from a state of control to a lesser state of control. A canonical use case is software bugs. Bugs that move from development to production have escaped. While this is a fairly common software measure, it has not been applied broadly to security. Chapter 7 focuses on escape rates.

Modern software development is predicated on speed of release. Being able to reduce escapes without impacting velocity would be an important asset in such environments. (Consider Spotify. It deploys upwards of 20,000 times a day.)[3]

There is a rate at which risk escapes. How might you measure that?



A reasonable place to start would be with naive escape rates. That is a simple cumulative ratio of vulnerabilities found in stage over those found in production. At first, there would not be a one-to-one correlation between vulnerabilities found in development to those later discovered in production, of course. The law of large numbers would soon produce stable rates. As correlative capabilities improve, one-to-one symmetry could be achieved. That is just not the place to start for 99% of teams.

Next Steps

This chapter sets both the ethos and direction for what is to come. The next section of the book focuses almost exclusively on the five BOOM baselines. Along the way, you will get exposed to Bayesian data analysis.

The goal is not a cookie-cutter approach to metrics. Rather, it is a model-based approach. The intent is to expose security practitioners to new ideas – to bring more into the measurement fold, and to help lift security up as a practice and career through better measurement.

Administrivia and Sundry Items

Book Site

All code will be hosted on the book's site: www.themetricsmanifesto.com. As is only natural for books with lots of code – there will be updates. Thus, I will publish bug fixes along with new and interesting tools with regularity.

Getting R

You can download R at https://cran.r-project.org/mirrors.html. CRAN stands for Comprehensive R Archive Network. Select the CRAN mirror site closest to you, then use the box labeled "Download and Install R." Or, you can use one of these two options based on your operating system:

- https://cloud.r-project.org/bin/macosx/
- https://cloud.r-project.org/bin/windows/

Getting RStudio

RStudio is an "integrated development environment," or IDE. While you likely only copy and paste code in this book, RStudio can be a convenience for running R applications. You can download and install it from the links found here:

- https://www.rstudio.com/products/rstudio/download/#download
  - Windows 10
  - Mac

Learning R

There are hundreds if not thousands of freely available resources for learning R. Here are a few you may want to consider. I frankly have not tried them. Google seemed to like them:

- Free Code Camp: https://www.youtube.com/watch?v=_V8eKsto3Ug
- Tutorial Point: https://www.tutorialspoint.com/r/index.htm
- Code Academy: https://www.codecademy.com/learn/learn-r

How I Learned R

My learning path on R started about 10 years ago. I used books, Google, and coffee. I rarely if ever read plain-ol'-coding books. My favorite materials are from people dealing with small and messy data sets. This includes research psychologists, cognitive scientists, evolutionary biologists, and such. Many of them are instructors who had to learn stats to do their jobs, as opposed to being statisticians already. Perhaps that makes them better explainers.

Notes

1. Slow Science Academy. (2021). The Slow Science Manifesto. http://slow-science.org/
2. Ellenberg, J. (2015). *How Not to Be Wrong: The Power of Mathematical Thinking* (Illustrated ed.). Penguin Books.
3. Nelson, D. (2020, November 20). Spotify scales its infrastructure with thousands of microservices, open source, and "fail faster" approach. SiliconANGLE. https://siliconangle.com/2020/11/20/spotify-scales-infrastructure-thousands-microservices-open-source-fail-faster-approach-kubecon/#:%7E:text=With%20thousands%20of%20data%20pipelines,today%2C%20if%20not%20more%20so