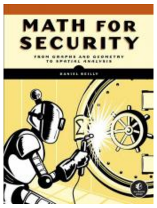
Review of [1]

Math for Security

From Graphs and Geometry to Spatial Analysis

Daniel Reilly

No Starch Press, 2023
292 pages, $49.99 Paperback, $39.99 Ebook



Review by
William Gasarch (gasarch@umd.edu)

1 Introduction

Seeing the title *Math for Security*, I first thought this book was about cryptography. Wrong! This book does not have any cryptography in it. Seeing the subtitle *From Graphs and Geometry to Spatial Analysis*, I first thought this book was about modeling networks as graphs. Right! Modeling networks as graphs is a major part of this book. My second thought was *Geometry?* Neither wrong nor right, since this is a question, not a guess on contents. We will discuss applications of geometry to security later in this review.

This is a very practical book for security. The math is introduced and used as needed. Reading it broke me out of my notion that the main use of math in security is cryptography.

The book is in three parts. Part I has 2 chapters, Part II has 8 chapters, and Part III has 3 chapters.

2 Summary of Contents

2.1 Part I: Environment and Conventions

Chapter 1 is aptly named *Setting up the Environment* and tells you how to install Python and some auxiliary packages onto your laptop. Chapter 2 is aptly named *Programming and Math Conventions* and goes over standard notation. Chapter 2 is indicative of the entire book in that they *do not* separate the math from the programming. They go hand in hand. This is most welcome.

2.2 Part II: Graph Theory and Computational Geometry

Chapter 3 is titled *Securing Networks with Graph Theory*. First it defines what a graph is and gives some examples. It then focuses on graphs where the nodes are either people or databases or systems.

The key question is *which node is the most important*, as that may be a node you want to attack (if you are a black hat) or protect (if you are a white hat). The chapter defines various notions of *importance* rigorously. The definition of *Betweenness Centrality* really intrigued me, since I had seen it before in a paper on 3SUM-hardness and did not know it had *real* applications.

Other questions the chapter considers are *which people are talking to each other*, which is essentially finding cliques, and connectivity of a network, which is just what you think it is.

Chapter 3 is mostly theoretical. Chapters 4 and 5, titled *Building a Network Traffic Analysis Tool* and *Identifying Threats with Social Network Analysis*, use the notions in Chapter 3 to actually build tools for security: programs that analyze traffic to find which nodes (processors or similar) are most heavily used, and identifying security threats. These chapters are *not* theoretical. They give actual code (in Python), and at the end the reader can actually build these tools.

Chapters 4 and 5 use historical data to analyze a network: they can be used to find a security threat *after its happened.* Chapter 6, titled *Analyzing Social Networks to Prevent Security Incidents*, is about predictive analysis— building tools to predict security threats *before they happen.* And again, this is not theoretical— at the end the reader can actually build these tools.

Chapters 3, 4, 5, 6 use graph theory for network analysis. Chapter 7 introduces geometry as a tool for security. Geometry? Chapter 7, titled *Using Geometry to Improve Security Packages*, gives the basics of computational geometry and some applications to planning a concert and to placing guards. Placing guards is security of the old-fashioned kind: security personnel who guard doors and such. And again, they give actual code (in Python), and at the end the reader can actually build these tools.

Chapters 8, 9, and 10, titled *Tracking People in Physical Space with Digital Information*, *Computational Geometry for Safety Resource Distribution*, and *Computational Geometry for Facial Recognition*, are about applying computational geometry to modern security concerns: tracking people (the ethics of this practice is discussed briefly), allocation of security resources, and facial recognition. Chapter 9 introduces Voronoi diagrams. This is typical of the book: math is developed on an as-needed basis. And again, they give actual code (in Python), and at the end the reader can actually build these tools.

2.3 Part III: The Art Gallery Problem

The basic Art Gallery problem is as follows:

Given an art gallery, represented by an n-sided polygon, what is the minimum number of guards needed to be placed in the polygon so every point in the polygon is visible to some guard.

In this form, this seems like a nice theoretical problem. And indeed, there has been a lot of work on it by computational geometers and other theorists. As usual, this book takes some of those algorithms, makes them practical, and again, they give actual code (in Python), and at the end the reader can actually build these tools.

But that's not all. This book takes the applications to security *very seriously.* In fact, the author came across the problem because it really was the security problem he was working on. Here is the quote from the book (p. 210):

A good plan for the placement of security personnel, checkpoints, and monitoring devices can reduce the number of incidents the security team will need to respond to from the start. It can also reduce the response time when an incident does occur, thus reducing the overall risk. Unfortunately, there are often differing levels of understanding among human planners on a security team, which can lead to poorly planned (or poorly implemented) security controls. That's why I am always searching for ways to automate portions of my team's planning.

It was during one of these searches that I discovered the Art Gallery problem, which addresses the very problem I was researching: the efficient deployment of security resources for buildings with what we'll call "untraditional layouts".

The formulation of the problem I gave above is not sufficient for real world usage. Consider the following issues that the formulation ignored:

1. In the real world, art galleries and other sensitive areas are a lot more complicated than n-sided polygons.

2. The guards have a limited range of angles their eyes can see.

3. The guards cannot honestly sing the song by *The Who* which goes: *I can see for miles and miles and miles and miles and miles.*

4. Some parts of the art gallery are more important than others to cover. (Be careful here. Some modern art looks like garbage, and some is literally made out of garbage but is valuable.)

5. Getting a fast algorithm is important.

Chapter 11, titled *Distributing Security Resources to Guard a Space*, defines the problem and begins to tackle some of the issues above. Chapters 12 and 13, titled *The Minimal Viable Product Approach to Security Software Development* and *Delivering Python Applications*, are about how to make the code into a package that people can actually use.

3 Opinion

This book was, for me, a real eye-opener. I am a theorist who is often skeptical about the problems theorists work on having any application. This book presents problems that have applications since:

1. The book *starts* with security needs and then sees what theory out there can help.

2. The book goes the extra n steps from a theoretical algorithm to an implementation.

Who should read this book? Both people in security and people in theory, as this book shows how theory can apply to security, and security can give new problems for theorists. The book can be read by an undergraduate who knows some discrete math and some basic programming.