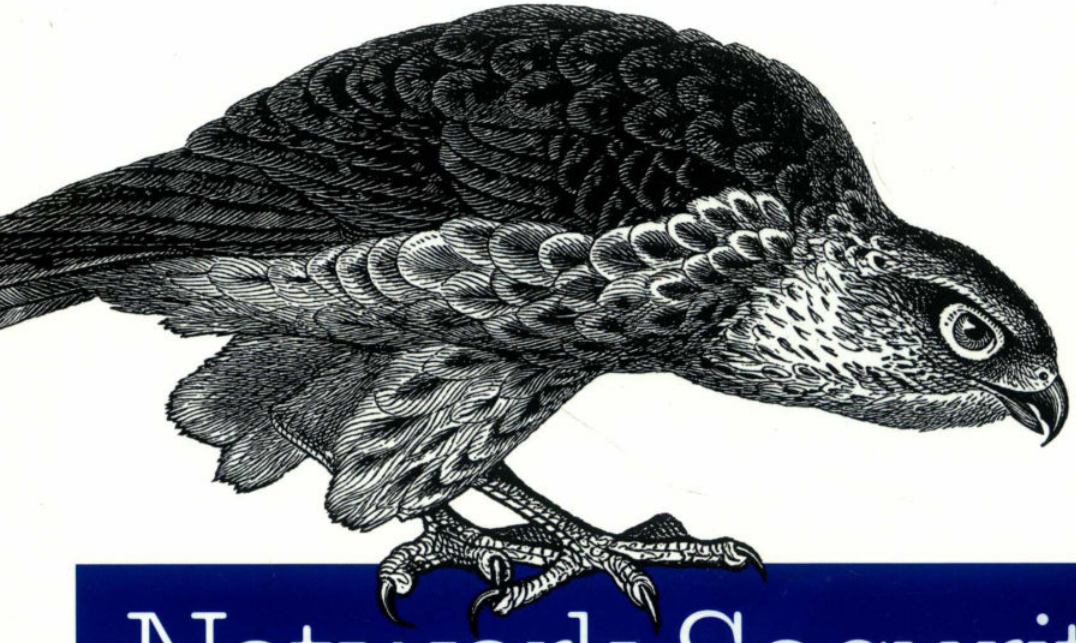
O'REILLY®

第2版



Network Security Through Data Analysis

基于数据分析的网络安全（影印版）

Michael Collins 著

東南大學出版社

第2版

基于数据分析的网络安全（影印版）

Network Security Through Data Analysis

Michael Collins 著

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

O'Reilly Media, Inc.授权东南大学出版社出版

南京 东南大学出版社

图书在版编目(CIP)数据

基于数据分析的网络安全:第2版:英文/(美)迈克尔·柯林斯(Michael Collins)著. —影印本. —南京:东南大学出版社,2018.7

书名原文:Network Security Through Data Analysis, 2E

ISBN 978-7-5641-7730-0

Ⅰ.①基… Ⅱ.①迈… Ⅲ.①计算机网络—网络安全—英文 Ⅳ.①TP393.08

中国版本图书馆CIP数据核字(2018)第090454号

图字:10-2018-109号

基于数据分析的网络安全 第2版(影印版)

出版发行:东南大学出版社

地 址:南京四牌楼2号 邮编:210096

出 版 人:江建中

网 址:http://www.seupress.com

电子邮件:press@seupress.com

印 刷:常州市武进第三印刷有限公司

开 本:787毫米×980毫米 16开本

印 张:26.75

字 数:524千字

版 次:2018年7月第1版

印 次:2018年7月第1次印刷

书 号:ISBN 978-7-5641-7730-0

定 价:99.00元

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

Praise for *Network Security Through Data Analysis, Second Edition*

Attackers generally know our technology better than we do, yet a defender's first reflex is usually to add more complexity, which just makes the understanding gap even wider—we won't win many battles that way. Observation is the cornerstone of knowledge, so we must instrument and characterize our infrastructure if we hope to detect anomalies and predict attacks. This book shows how and explains why to observe that which we defend, and ought to be required reading for all SecOps teams.

—Dr. Paul Vixie, CEO of Farsight Security

Michael Collins provides a comprehensive blueprint for where to look, what to look for, and how to process a diverse array of data to help defend your organization and detect/deter attackers. It is a "must have" for any data-driven cybersecurity program.

—Bob Rudis, Chief Data Scientist, Rapid7

Combining practical experience, scientific discipline, and a solid understanding of both the technical and policy implications of security, this book is essential reading for all network operators and analysts. Anyone who needs to influence and support decision making, both for security operations and at a policy level, should read this.

—Yurie Ito, Founder and Executive Director, CyberGreen Institute

Michael Collins brings together years of operational expertise and research experience to help network administrators and security analysts extract actionable signals amidst the noise in network logs. Collins does a great job of combining the theory of data analysis and the practice of applying it in security contexts using real-world scenarios and code.

—Vyas Sekar, Associate Professor, Carnegie Mellon University/CyLab

Preface

This book is about networks: monitoring them, studying them, and using the results of those studies to improve them. "Improve" in this context hopefully means to make more secure, but I don't believe we have the vocabulary or knowledge to say that confidently—at least not yet. In order to implement security, we must know what decisions we can make to do so, which ones are most effective to apply, and the impact that those decisions will have on our users. Underpinning these decisions is a need for *situational awareness.*

Situational awareness, a term largely used in military circles, is exactly what it says on the tin: an understanding of the environment you're operating in. For our purposes, situational awareness encompasses understanding the components that make up your network and how those components are used. This awareness is often *radically* different from how the network is configured and how the network was originally designed.

To understand the importance of situational awareness in information security, I want you to think about your home, and I want you to count the number of web servers in your house. Did you include your wireless router? Your cable modem? Your printer? Did you consider the web interface to CUPS? How about your television set?

To many IT managers, several of the devices just listed won't have registered as "web servers." However, most modern embedded devices have dropped specialized control protocols in favor of a web interface—to an outside observer, they're just web servers, with known web server vulnerabilities. Attackers will often hit embedded systems without realizing what they are—the SCADA system is a Windows server with a couple of funny additional directories, and the MRI machine is a perfectly serviceable spambot.

This was all an issue when I wrote the first edition of the book; at the time, we discussed the risks of unpatched smart televisions and vulnerabilities in teleconferencing systems. Since that time, the Internet of Things (IoT) has become even more of a

thing, with millions of remotely accessible embedded devices using simple (and insecure) web interfaces.

This book is about collecting data and looking at networks in order to understand how the network is used. The focus is on analysis, which is the process of taking security data and using it to make actionable decisions. I emphasize the word *actionable* here because effectively, security decisions are restrictions on behavior. Security policy involves telling people what they shouldn't do (or, more onerously, telling people what they *must* do). Don't use a public file sharing service to hold company data, don't use *123456* as the password, and don't copy the entire project server and sell it to the competition. When we make security decisions, we interfere with how people work, and we'd better have good, solid reasons for doing so.

All security systems ultimately depend on users recognizing and accepting the tradeoffs—inconvenience in exchange for safety—but there are limits to both. Security rests on people: it rests on the individual users of a system obeying the rules, and it rests on analysts and monitors identifying when rules are broken. Security is only marginally a technical problem—information security involves endlessly creative people figuring out new ways to abuse technology, and against this constantly changing threat profile, you need cooperation from both your defenders and your users. Bad security policy will result in users increasingly evading detection in order to get their jobs done or just to blow off steam, and that adds additional work for your defenders.

The emphasis on actionability and the goal of achieving security is what differentiates this book from a more general text on data science. The section on analysis proper covers statistical and data analysis techniques borrowed from multiple other disciplines, but the overall focus is on understanding the structure of a network and the decisions that can be made to protect it. To that end, I have abridged the theory as much as possible, and have also focused on mechanisms for identifying abusive behavior. Security analysis has the unique problem that the targets of observation are not only aware they're being watched, but are actively interested in stopping it if at all possible.

The MRI and the General's Laptop

Several years ago, I talked with an analyst who focused primarily on a university hospital. He informed me that the most commonly occupied machine on his network was the MRI. In retrospect, this is easy to understand.

"Think about it," he told me. "It's medical hardware, which means it's certified to use a specific version of Windows. So every week, somebody hits it with an exploit, roots it, and installs a bot on it. Spam usually starts around Wednesday." When I asked why he didn't just block the machine from the internet, he shrugged and told me the doctors

wanted their scans. He was the first analyst I'd encountered with this problem, but he wasn't the last.

We see this problem a lot in any organization with strong hierarchical figures: doctors, senior partners, generals. You can build as many protections as you want, but if the general wants to borrow the laptop over the weekend and let his granddaughter play Neopets, you've got an infected laptop to fix on Monday.

I am a firm believer that the most effective way to defend networks is to secure and defend *only* what you need to secure and defend. I believe this is the case because information security will always require people to be involved in monitoring and investigation—the attacks change too frequently, and when we automate defenses, attackers figure out how to use them against us.[1]

I am convinced that security should be inconvenient, well defined, and constrained. Security should be an artificial behavior extended to assets that must be protected. It should be an artificial behavior because the final line of defense in any secure system is the *people* in the system—and people who are fully engaged in security will be mistrustful, paranoid, and looking for suspicious behavior. This is not a happy way to live, so in order to make life bearable, we have to limit security to what must be protected. By trying to watch everything, you lose the edge that helps you protect what's really important.

Because security is inconvenient, effective security analysts must be able to *convince* people that they need to change their normal operations, jump through hoops, and otherwise constrain their mission in order to prevent an abstract future attack from happening. To that end, the analysts must be able to identify the decision, produce information to back it up, and demonstrate the risk to their audience.

The process of data analysis, as described in this book, is focused on developing security knowledge in order to make effective security decisions. These decisions can be forensic: reconstructing events after the fact in order to determine why an attack happened, how it succeeded, or what damage was done. These decisions can also be proactive: developing rate limiters, intrusion detection systems (IDSs), or policies that can limit the impact of an attacker on a network.

Audience

The target audience for this book is network administrators and operational security analysts, the personnel who work on NOC floors or who face an IDS console on a

1 Consider automatically locking out accounts after *x* number of failed password attempts, and combine it with logins based on email addresses. Consider how many accounts an attacker can lock out that way.

regular basis. Information security analysis is a young discipline, and there really is no well-defined body of knowledge I can point to and say, "Know this." This book is intended to provide a snapshot of analytic techniques that I or other people have thrown at the wall over the past 10 years and seen stick. My expectation is that you have some familiarity with TCP/IP tools such as `netstat`, `tcpdump`, and `wireshark`.

In addition, I expect that you have some familiarity with scripting languages. In this book, I use Python as my go-to language for combining tools. The Python code is illustrative and might be understandable without a Python background, but it is assumed that you possess the skills to create filters or other tools in the language of your choice.

In the course of writing this book, I have incorporated techniques from a number of different disciplines. Where possible, I've included references back to original sources so that you can look through that material and find other approaches. Many of these techniques involve mathematical or statistical reasoning that I have intentionally kept at a functional level rather than going through the derivations of the approach. A basic understanding of statistics will, however, be helpful.

Contents of This Book

This book is divided into three sections: Data, Tools, and Analytics. The Data section discusses the process of collecting and organizing data. The Tools section discusses a number of different tools to support analytical processes. The Analytics section discusses different analytic scenarios and techniques. Here's a bit more detail on what you'll find in each.

Part I discusses the collection, storage, and organization of data. Data storage and logistics are critical problems in security analysis; it's easy to collect data, but hard to search through it and find actual phenomena. Data has a footprint, and it's possible to collect so much data that you can never meaningfully search through it. This section is divided into the following chapters:

Chapter 1
: This chapter discusses the general process of collecting data. It provides a framework for exploring how different sensors collect and report information and how they interact with each other, and how the process of data collection affects the data collected and the inferences made.

Chapter 2
: This chapter expands on the discussion in the previous chapter by focusing on sensor placement in networks. This includes points about how packets are transferred around a network and the impact on collecting these packets, and how various types of common network hardware affect data collection.

Chapter 3

This chapter focuses on the data collected by network sensors including `tcpdump` and NetFlow. This data provides a comprehensive view of network activity, but is often hard to interpret because of difficulties in reconstructing network traffic.

Chapter 4

This chapter focuses on the process of data collection in the service domain—the location of service log data, expected formats, and unique challenges in processing and managing service data.

Chapter 5

This chapter focuses on the data collected by service sensors and provides examples of logfile formats for major services, particularly HTTP.

Chapter 6

This chapter discusses host-based data such as memory and disk information. Given the operating system–specific requirements of host data, this is a high-level overview.

Chapter 7

This chapter discusses data in the active domain, covering topics such as scanning hosts and creating web crawlers and other tools to probe a network's assets to find more information.

Part II discusses a number of different tools to use for analysis, visualization, and reporting. The tools described in this section are referenced extensively in the third section of the book when discussing how to conduct different analytics. There are three chapters on tools:

Chapter 8

This chapter is a high-level discussion of how to collect and analyze security data, and the type of infrastructure that should be put in place between sensor and SIM.

Chapter 9

The System for Internet-Level Knowledge (SiLK) is a flow analysis toolkit developed by Carnegie Mellon's CERT Division. This chapter discusses SiLK and how to use the tools to analyze NetFlow, IPFIX, and similar data.

Chapter 10

One of the more common and frustrating tasks in analysis is figuring out where an IP address comes from. This chapter focuses on tools and investigation methods that can be used to identify the ownership and provenance of addresses, names, and other tags from network traffic.

Part III introduces analysis proper, covering how to apply the tools discussed throughout the rest of the book to address various security tasks. The majority of this section is composed of chapters on various constructs (graphs, distance metrics) and security problems (DDoS, fumbling):

Chapter 11

Exploratory data analysis (EDA) is the process of examining data in order to identify structure or unusual phenomena. Both attacks and networks are moving targets, so EDA is a necessary skill for any analyst. This chapter provides a grounding in the basic visualization and mathematical techniques used to explore data.

Chapter 12

Log data, payload data—all of it is likely to include some forms of text. This chapter focuses on the encoding and analysis of semistructured text data.

Chapter 13

This chapter looks at mistakes in communications and how those mistakes can be used to identify phenomena such as scanning.

Chapter 14

This chapter discusses analyses that can be done by examining traffic volume and traffic behavior over time. This includes attacks such as DDoS and database raids, as well as the impact of the workday on traffic volumes and mechanisms to filter traffic volumes to produce more effective analyses.

Chapter 15

This chapter discusses the conversion of network traffic into graph data and the use of graphs to identify significant structures in networks. Graph attributes such as centrality can be used to identify significant hosts or aberrant behavior.

Chapter 16

This chapter discusses the unique problems involving insider threat data analysis. For network security personnel, insider threat investigations often require collecting and comparing data from a diverse and usually poorly maintained set of data sources. Understanding what to find and what's relevant is critical to handling this trying process.

Chapter 17

Threat intelligence supports analysis by providing complementary and contextual information to alert data. However, there is a plethora of threat intelligence available, of varying quality. This chapter discusses how to acquire threat intelligence, vet it, and incorporate it into operational analysis.

Chapter 19
: This chapter discusses a step-by-step process for inventorying a network and identifying significant hosts within that network. Network mapping and inventory are critical steps in information security and should be done on a regular basis.

Chapter 20
: Operational security is stressful and time-consuming; this chapter discusses how analysis teams can interact with operational teams to develop useful defenses and analysis techniques.

Changes Between Editions

The second edition of this book takes cues from the feedback I've received from the first edition and the changes that have occurred in security since the time I wrote it. For readers of the first edition, I expect you'll find about a third of the material is new. These are the most significant changes:

- I have removed R from the examples, and am now using Python (and the Anaconda stack) exclusively. Since the previous edition, Python has acquired significant and mature data analysis tools. This also saves space on language tutorials which can be spent on analytics discussions.
- The discussions of host and active domain data have been expanded, with a specific focus on the information that a network security analyst needs. Much of the previous IDS material has been moved into those chapters.
- I have added new chapters on several topics, including text analysis, insider threat, and interacting with operational communities.

Most of the new material is based around the idea of an analysis team that interacts with and supports the operations team. Ideally, the analysis team has some degree of separation from operational workflow in order to focus on longer-term and larger issues such as tools support, data management, and optimization.

Tools of the Trade

So, given Python, R, and Excel, what should you learn? If you expect to focus purely on statistical and numerical analysis, or you work heavily with statisticians, learn R first. If you expect to integrate tightly with external data sources, use techniques that aren't available in CRAN, or expect to do something like direct packet manipulation or server integration, learn Python (ideally iPython and Pandas) first. Then learn Excel, *whether you want to or not*. Once you've learned Excel, take a nice vacation and then learn whatever tool is left of these three.

All of these data analysis environments provide common tools: some equivalent of a data frame, visualization, and statistical functionality. Of the three, the Pandas stack (that is, Python, NumPy, SciPy, Matplotlib, and supplements) provides the greatest variety of tools, and if you're looking for something outside of the statistical domain, Python is going to have it. R, in comparison, is a tightly integrated statistical package where you will always find the latest statistical analysis and machine learning tools. The Pandas stack involves combining multiple toolsets developed in parallel, resulting in both redundancy and valuable tools located all over the place. R, on the other hand, inherits from this parallel development community (via S and SAS) and sits in the developer equivalent of the Uncanny Valley.

So why Excel? Because operational analysts live and die off of Excel spreadsheets. Excel integration (even if it's just creating a button to download a CSV of your results) will make your work relevant to the operational floor. Maybe you do all your work in Python, but at the end, if you want analysts to use it, give them something they can plunk into a spreadsheet.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords. Also used for commands and command-line utilities, switches, and options.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/mpcollins/nsda_examples*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of

the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Network Security Through Data Analysis* by Michael Collins (O'Reilly). Copyright 2017 Michael Collins, 978-1-491-96284-8."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Safari

Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit *http://oreilly.com/safari*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/nstda2e*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

I need to thank my editors, Courtney Allen, Virginia Wilson, and Maureen Spencer, for their incredible support and feedback, without which I would still be rewriting commentary on regression over and over again. I also want to thank my assistant editors, Allyson MacDonald and Maria Gulick, for riding herd and making me get the thing finished. I also need to thank my technical reviewers: Markus DeShon, André DiMino, and Eugene Libster. Their comments helped me to rip out more fluff and focus on the important issues.

This book is an attempt to distill down a lot of experience on ops floors and in research labs, and I owe a debt to many people on both sides of the world. In no particular order, this includes Jeff Janies, Jeff Wiley, Brian Satira, Tom Longstaff, Jay Kadane, Mike Reiter, John McHugh, Carrie Gates, Tim Shimeall, Markus DeShon, Jim Downey, Will Franklin, Sandy Parris, Sean McAllister, Greg Virgin, Vyas Sekar, Scott Coull, and Mike Witt.

Finally, I want to thank my mother, Catherine Collins.

Table of Contents