SYNGRESS

BREAKING INTO INFORMATION SECURITY

CRAFTING A CUSTOM CAREER PATH TO GET THE JOB YOU REALLY WANT

Josh More | Anthony J. Stieber | Chris Liu

Breaking into Information Security

Crafting a Custom Career Path to Get the Job You Really Want

Josh More

Anthony J. Stieber

Chris Liu

Technical Editor: Beth Friedman

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Syngress is an Imprint of Elsevier

SYNGRESS.

Acquiring Editor: Chris Katsaropoulos
Editorial Project Manager: Anna Valutkevich
Project Manager: Priya Kumaraguruparan
Designer: Matthew Limbert

Syngress is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-800783-9

Contents

Author Biographies

JOSH MORE

Josh More started Eyra Security after spending more than 15 years in IT. He holds multiple security and technical certifications and serves in a leadership position on several security-focused groups. When taking a break from reducing IT and security risks for his company's customers, Josh enjoys reading, cooking, and photography.

ANTHONY J. STIEBER

Anthony J. Stieber has spent over 20 years in academia, banks, retail, information security, and insurance; designed enterprise security architectures; installed military and commercial firewalls; engineered medical diagnostic systems; reverse-engineered Internet stores; encrypted data warehouses; provided expertise for legal cases; spoken at international cryptography conferences; broken encrypted storage systems; studied as an apprentice locksmith; and became a published writer and recently a book co-author.

CHRIS LIU

Chris Liu has over 20 years of information technology experience, a CISSP, CISM, and no idea how he ended up where is. He has been a help desk technician, network administrator, quality assurance engineer, release manager, IT manager, instructor, developer, consultant, and product development manager, and is currently an information security professional. He is proof that careers sometimes only make sense in retrospect.

Page left intentionally blank

Acknowledgments

Many people helped us with the book; family and friends provided support and understanding for the many nights, evenings, and weekends we each worked on the book.

We are particularly grateful to all of those who contributed the information security stories in the book, and regret those we couldn't include:

Travis Abrams, Jim Chan, David Henning, Michael Huber, Heather Kohls, Mak Kolybabi, Joshua Marpet, John Meyers, Stephen Northcutt, Greg Sullivan, Jimmy Vo, Alan Waggoner

We support free and open-source software and its communities and believe it is critical to information security. GNU Emacs, GNU/Linux Knoppix, LibreOffice, Mozilla Firefox, Perl, VIM, and much other free open-source software were used in the production of this book.

Page left intentionally blank

CHAPTER

INTRODUCTION 0.1

INTRODUCTION

This book was written by three people with three vastly different experiences in information security. The book has been influenced by everyone with whom we've worked and by every book and article we've read. So this book is from the information security community. This book has illustrative stories of over a dozen people, describing their own information security stories. The number of people involved in this book is too many to count and goes well beyond those listed on the cover and in the Acknowledgments.

Information security is constantly changing, and we expect this book will also change to keep pace. The second edition will involve even more people and cover even more topics. We don't know what the third or even fourth editions will cover.

It is our aim that this book will grow, not just with our own careers, but yours as well. With that in mind, as you read this book, please feel free to tell us what has been important to you so we can include it in the next edition.

As authors of a community book, we feel that it is important that the book not only be from the community but also be part of the community. As such, we have earmarked a portion of the royalties of this book to be donated to the Hackers for Charity nonprofit organization.

WHO SHOULD READ THIS BOOK

This book is for anyone changing roles into or within the security community. While it will likely be of more interest to people trying to break into entry level information security, the book is written so that you may break into any role, not just at the beginning of your career. Whether you're just getting started as a security analyst or are becoming a penetration testing lead in charge of your team, there should be something in this book of interest to you.

HOW TO READ THIS BOOK

This book is a survey of the information security job market and community, not a direct path to success. Information security and technology changes quickly, so any direct advice given will quickly go out-of-date. Instead, we propose a different way of thinking about your career.

Careers often follow a path of three phases or "tiers" in which you first spend most of your time learning, then spend a large amount of your time doing what you've learned, and then you may focus

on teaching others. This book follows these three tiers with a Learn/Do/Teach approach. While any one role will likely involve all three tiers, the proportions of Learn/Do/Teach will change as you progress.

To read this book, read the "core" of each first, going through Models, then Learn, then Do, then Teach. Each has descriptions of several information security roles. Feel free to jump around and read what the different roles involve. Once you know what work you want to do, consider which roles earlier in the process appeal to you the most. This should help you to create your own custom career path, which will both be more rewarding and more likely to succeed than anything any of us could lay out for you. For many people, career paths are dynamic, and change as roles or jobs change.

As you progress through your information security journey, keep your goals in mind, but also keep in mind that your goals may change. Both your goals and your environment determine the path your career will take. It may be that this path will not lead you to your new goals, so pay attention as things change and adjust accordingly.

CAUTION

Ethics and Career

The cautions in this book aren't just because you could be responsible for lost or damaged data, but also because you could lose your job and damage your career. You are responsible for your life, and your career is part of your life. You are responsible for your own career, and you are responsible for using your own judgment. Ethics matter in information security; any poor ethics and bad judgment will make a lot of trouble for yourself and others in your current or future jobs.

There is no guarantee that the path you choose will work for you, so if you find yourself at a dead end, consider other options. If you keep building your skills and remain persistent, you can get where you want to go.

This book provides a framework for thinking about your career. Careers move forward in fits and starts, so be prepared to fail fast, recover fast, and start over in another role as you move to where you want to go. This book is not a hard-and-fast guide, rather it is a steady and slow career guide. Your career will probably not be like any of ours, or anyone else's. If you see a path we didn't define, consider it, if it works for you, or doesn't, share it with the community and us. As the stories in the book show, there is no single true path to success.

NOTES FROM THE AUTHORS

We are three authors attempting to speak with one voice. It was not always possible; but where we had conflicts, we worked them out together. But there is also value in us each speaking individually.

WHY COMMUNITY? — JOSH MORE

Information security is a losing game. Our adversaries — the attackers — are better-funded than we defenders, and they have more time to cause problems than we have to fix them. This will not change. Throughout history, the cycle of attack and defense arms race has been built on the premise of "good

enough." A wooden shield is a good enough defense until your enemies get metal lances and longbows. A stone castle wall is a good enough defense until your enemies get siege engines (or helicopters).

The fact is, someone is always going to lose because once we get to war, it's too late for a win-win situation. And with every win, the attackers are going to get a little bit better. They are tuning their tools and techniques every day, while far too many of us defenders are spending our time just catching-up. In order to survive, we have to learn as quickly as they do, and the only way to do that is to share knowledge.

This is what community is about. Our community is not perfect; but we are getting better at sharing. When I started, companies were loath to admit that they had experienced an attack, much less were breached. Today, we're seeing reports of major data breaches monthly. The more we talk about what everybody faces, the better we can work together. Knowing what we're thinking does give attackers an edge. However, on balance, the boost we get from sharing knowledge is greater than the increase in our risk.

And really, that's what it is about. As an information security professional, your job is about balancing risk. However, you will almost never be the sole decision maker. You will explain the risks as you see them and you'll have to understand those who see them differently. To win these battles and increase your chances of surviving the constant war, you'll need help. After you read this book, talk about your ideas with your local security groups, on mailing lists and blogs, and at conferences. Seek out those who disagree with you and learn how they think. Give feedback, so we can all improve.

This even applies to book authors. This book is written by three people with collectively over 50 years of experience in the industry. But still, we're just three. We've asked for help from a handful of others, but we're also asking for help from you. If you help us help others, we all get better. If we get better faster than the attackers, we can improve everyone's defense.

SECURE THINKING? — ANTHONY J. STIEBER

The biggest difference I have seen in being good at security, not just information security, is an attitude, a mindset, to think in ways that others don't. This isn't about being smart, imaginative, educated, technically skilled, or experienced, although those can help. I have met too many smart, imaginative, educated, technically skilled, and experienced people who can't imagine security problems. They are neither stupid nor ignorant, and they are very good at other things; but they aren't very good at security. Unfortunately, some of them are in the security industry.

For example, it doesn't occur to them that their system will be attacked by someone as smart, educated, and experienced as themselves. Perhaps this is an innate goodness in them, or a lack of empathy for someone else's goals. Successful defending means being able to think at least a little like the attacker, ideally before the attacker does. This doesn't require superhuman thinking, the ability to predict the future, or being a bad person—it just means thinking enough like an attacker *before* getting attacked.

If you can think about what an attacker might do at the same time you are trying to defend, you'll be better at security than those who can't.

Some defenders, such as security researchers and penetration testers, go further and even act like attackers. If you can think about two different and incompatible ideas at the same time, if you can ask that next question, if you have the empathy to think like an attacker, but have the sympathy to not be an attacker, then you can break into information security. Everything else you can learn, and teach others so they can do security better. If you can do this, then security could be right for you.

Empathize with your adversaries, and defeat them anyway.

IS SECURITY RIGHT FOR ME? — CHRIS LIU

As an instructor who has taught security both to college students and professionals, I have found that many people are interested in being information security professionals. Unfortunately, not as many people are interested in learning *how* to be a security professional. What do I mean by this? Simple: There are no shortcuts. You must get down and dirty with technical information. You need to become intimately familiar with bits and bytes that are boring and challenging at the same time. You need to be comfortable—very comfortable—with things not working the way you expect.

I have generally been able to spot those who will do well with security by the presence of a single attitude. Do they want to learn as much as they possibly can? Are they willing to explore stray paths and dead ends, but use those to learn from their mistakes? Or do they ask the question that gives it all away: "Do I need to know this for the test?"

Yes, hacking is cool. Being able to attack websites is neat. But being able to actually analyze a disparate set of data, and develop a cohesive vision of the target takes time, patience, and the ability to think outside the box. If a probe gives you an unexpected response, you need to be able to analyze that information and use it to create a new probe. If you can only run by the script, you will never get to the cool stuff.

Things are always going wrong when you are doing security. The script that worked the last time to attack a web server doesn't work this time, even though it should. Well, you think it should. But you weren't aware that this new client had a slightly different configuration that made this attack entirely irrelevant. Are you able to look at a long list of failed attempts and go, "Well, at least I know this won't work here," and develop a new strategy?

Are you able to learn, while you are doing? Are you able to teach while you are learning? If you can, then security could be the right fit for you.

If not, you may discover that security is more frustrating than cool for you.

TERMINOLOGY

Cyber, Hacking, and Information Security Growing Pains

Information security is a young and immature field, even the term "discipline" can't really be applied yet, and the term "profession" is still debatable. Just being paid to do something doesn't make it a profession, it also has to get done properly, and right now information security is often not even done. Information security also has many common terms without commonly accepted meanings or are highly ambiguous. Some terms and some meanings are even controversial. The meanings of ordinary words like "defect," "exploit," "threat," "vulnerability," and "weakness," are still argued about. Some words are particularly controversial and are information security sub-culture shibboleths that will mark the speaker. For example, within some groups the word "hacker" means "computer criminal," in other groups it means "computer genius," and in other groups it means both.

Another common word is "cyber" and may mean "computers, the Internet, and command and control systems in general" or it can mean "I don't know I'm ignorant about computers or security". Cyber can also mean almost nothing.

To avoid the ambiguity of these words and others we've avoided them, except when used by others in context.

CHAPTER

MODELS 0.2

MODELS

> "Essentially, all models are wrong, but some are useful."
>
> **— George E. P. Box**

Humanity has acquired more knowledge than can fit in a single human brain. To help us understand what's going on, we continuously abstract concepts into other concepts. For example, most people don't need to know the differences between an Adirondack, a Bofinger, and a caquetoire. For everyday life, the abstract concept of "chair" will suffice. We do the same in information security. Networking is abstracted with the seven-layer OSI model, the four-layer TCP/IP model, or just a single "is it working?" layer on which other even higher layers are placed.

The point of a model is to simplify the world and make it more understandable. Albert Einstein is often quoted as saying "Make it as simple as possible, but no simpler." What Einstein actually wrote in the journal *Philosophy of Science* was:

> "It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience."

The fact that Einstein chose to keep his lesson on simplification rather complex indicates the type of people who typically read the journal *Philosophy of Science* in the early 01930s. This book is aimed at a somewhat different crowd. We will primarily use the model: Learn/Do/Teach.

TIME MANAGEMENT

Five Digit Years

As this book is about time management and taking the long term view, we have adopted the practice of The Long Now Foundation of writing years with five-digit dates. While we do not realistically believe that much of this book will be applicable past the year 09999 — information security changes rather quickly, after all — we do feel that deliberately thinking in a longer term that most people are used to will help you to realize the importance of taking the long view as you plan your information security career.

NOTE

References and URLs

Full references for many items mentioned in this book, such as Albert Einstein's article in Philosophy of Science, can be found in the "Appendix: People & Quotes". Others references are in broad categories in roughly the order as presented in the book, such as Security Models and Time Management.

LEARN/DO/TEACH

The Learn/Do/Teach model is adapted from the medical community. The core idea is that you learn significantly better if, after the initial learning, you actively apply it. Then, after you've demonstrated understanding, you teach someone else. This gives you the opportunity to learn without causing harm, and provides a chain of knowledge stretching from generation to generation.

This book is organized similarly. The first section, Learn, is about the importance of learning and lists common entry-level information security jobs in which you typically learn the basics. Entry-level jobs are seldom fun, but only by using them to acquire a firm background can you expect to gain the experience needed to excel at higher levels. To use your experience from other jobs in information security you can perform "lateral" moves into the Do and Teach sections of the model.

The Do section of the model introduces the importance of getting your hands dirty, literally and figuratively. It lists common mid-career jobs and discusses how there can be gaps between formal "book learning" and "the school of hard knocks" on the job. We do not prefer one option over the other but those who understand the theoretical underpinnings and then also get deep experience tend to be more successful. The point is to maximize the amount of skill you gain in a specific amount of time. Thus, if you first Learn, then Do, you can gain a significant amount of demonstrable skill. These are the roles on which a business succeeds or fails, and that put you at the heart of information security. Many people are happy working these sorts of jobs for their entire career.

The Teach section brings it together. To close the loop and pass the learning to the next generation, you give back to the community through teaching. Teaching helps solidify your thinking and makes learning new things more efficient. As with the Do section, there are lateral paths to jump into the Teach of information security. The jobs listed in the Teach section include commonly accepted senior level jobs. However, what distinguishes Teach from Do is that each of these jobs has a teaching component where you are expected to help your colleagues improve over time.

Finally, there is a section on Boosting. This section is separate because it is optional, but has highly recommended suggestions for bootstrapping skills on your own time. Boosting defeats the "experience needed to get experience" trap and shows how to use the Learn/Do/Teach cycle outside of work to rapidly gain the skills needed to make that leap to your desired job.

INFORMATION SECURITY MODELS

Other information security models are also used in this book. These are not core such as Learn/Do/Teach but they are commonly used in information security.

(ISC)^2 COMMON BODY OF KNOWLEDGE DOMAINS

The International Information Systems Security Certification Consortium or (ISC)^2 is best known for its stewardship of the Certified Information Security Systems Professional (CISSP) certification. This certification tests on what (ISC)^2 calls the CISSP Common Body of Knowledge (CBK). The CBK is organized into domains which are "buckets" of information security knowledge concentrations which form the (ISC)^2 view of a well-rounded security practitioner. In 02015 the ten domains were reorganized into eight domains but with the same information: Security and Risk Management, Asset Security, Security Engineering, Communication and Network Security, Identity and Access Management, Security Assessment and Testing, Security Operations, Software Development Security. See the Appendix: Models for details.

The CISSP CBK approach involves several domains of study that may not necessarily apply to your specific job or interests. As such, this approach may be considered too broad for some individuals. The CISSP is generally considered "broad but shallow" which is well suited for generalists who can consult specialists, but not as well suited for those same specialists.

CIA TRIAD

A simple approach to information security is the CIA triad which has the three qualities of confidentiality, integrity and availability. Sometimes this is called ACI to avoid confusion.

- **Confidentiality**—Is the quality that only those who need access to or knowledge of the data will have either. Many reports you see of people intruding on networks involves a loss or breakdown of confidentiality. Lost laptops, poor database and web site access controls, insecure email, and weak cryptography are common ways to lose confidentiality. Examples include medical record and credit card exposure. Common terms for confidentiality loss include data breach, data theft, exposure, leak, and piracy.
- **Integrity**—Is the quality of how trustworthy the system or the data within the system is—in other words, how confident you are that the system will respond as it should and that it has not been tampered with. No access controls, poor web site access controls, and weak cryptography are common ways to lose integrity. Examples include incomplete or wrong medical records and ATM skimmers. Common terms for integrity loss include data corruption, defacement, modification, subversion and tampering.
- **Availability**—As you might expect, describes how available the system is. This idea comes from such fields as health care, where unavailable information, such as a heart-rate monitor, may result in someone's death, or financial systems, where lost transaction data means lost money. No data backups, equipment failure, no redundancy, and unexpected load are common ways to lose availability. Hard disk drive crashes, failed backups, power outage, and very high and aggressively low popularity are common ways to lose availability. Common terms for availability loss include business continuity loss, data destruction or loss, outage and denial-of-service. A denial-of-service (DoS) attack is specifically against availability by an adversary. Good security detects, prevents, and withstands DoS attacks. Conversely, if a security feature protects a system by preventing anyone from accessing

it, it creates a low availability regardless of the "good" or "bad" status of the individuals using it.

All of the CIA triad qualities can be adversely affected by merely accident and natural disasters, not just deliberate actions by an adversary. This does not lessen their status as information security concerns. In addition adversaries can take advantage of accidents and disasters and make them worse.

It is useful to rank the three CIA triad qualities in order of importance which will vary. Some environments have mostly equally balanced requirements, but others will be biased toward or away from one or two.

The CIA triad approach makes some security issues very simple to explain and helps determine which quality to prioritize. Health care often prioritizes availability, finance requires integrity, and defense organizations tend to focus on confidentiality. The CIA triad is a powerful tool for thinking about information security, but like all models, it is limited.

PARKERIAN HEXAD

In 01998, Donn B. Parker proposed a different way of looking at information security. This approach doubles the number of qualities from the CIA triad, thus squaring the number of possibilities, but it is not as difficult to work with as the eight domains of the (ISC)2 Common Body of Knowledge (CBK). The variables are:

- **Confidentiality**—As in the CIA triad.
- **Possession or Control**—Covers situations where a data element or system escapes the scope of controls placed around it. It may or may not involve a compromise of Confidentiality, but simply knowing that it has escaped allows you to treat it as "tainted" until verified as unaccessed or altered.
- **Integrity**—As described in the CIA triad.
- **Authenticity**—Covers the creation of data or systems. Possession or Control focuses on potential losses of Confidentiality, but Authenticity focuses on potential losses of Integrity. If you prove a trusted system as being authentic, it may be necessary to create additional controls around it.
- **Availability**—As described in the CIA triad.
- **Utility**—To disclose a bias on the part of the authors, Utility is a critical aspect to security that most models miss. Utility combines the concepts of usefulness and usability. Some security features fail "closed," making it difficult or impossible for anyone to access the system. In such scenarios, Utility is zero and this can overwhelm any other issues at play, although it may very well be secure in that it is inaccessible. Utility can also be considered a measure of usability. Some security controls are extremely restrictive and users actively work against them to get their work done. Since security involves a mix of both technology and people, building an understanding of how people respond is critical to your model. An unusable system won't be used and security that isn't used isn't secure at all.

SANS LEARNING FAMILIES

SANS is one of the best security training companies in the world, and offers dozens of different training classes. As with many academic approaches to learning, SANS groups the classes into categories. This approach is similar to the families discussed earlier. As of October 02015, the classes break down into several categories with some overlap between each:

- Cyber Defense—general information security covering the basics and and more advanced use of common defensive tools such as firewalls, anti-virus, monitoring, auditing, and system hardening.
- System Administration—focuses on information security for the operating system administrator.
- Digital Forensic Investigations and Media Exploitation—catching concerns after they've been exploited. This can involve reverse-engineering malware and carving memory and disk to obtain evidence. In short, forensics focuses on determining what happened, so appropriate plans can be made to address any issues.
- Penetration Testing—using and creating tools to break into live wired and wireless networks, web applications, mobile devices.
- Incident Response—often combined with Digital Forensics to form DFIR, this is near-real-time and real-time response to attack using forensics, reverse engineering, and any and all of the other tools available in the other categories.
- Management—for people who are responsible for the business as a whole. This category of security learning involves understanding things at a very high level, so appropriate decisions can be made.
- Secure Software Development—all aspects of development, regardless of language. Many developers have a different approach to security, and this category is aimed at catching security issues earlier, as the sooner you find an issue, the cheaper it is to fix.
- Intrusion Analysis—after the fact analysis including defensive techniques used elsewhere and forensics.
- Cyber Guardian—a high level subset of Cyber Defense.
- Audit and Legal—focus on specific issues around standards, regulations, and other things that people are required to do from a security perspective so people can demonstrate compliance.
- Industrial Control Systems—industrial control systems (ICS) and supervisory, control and data acquisition (SCADA) systems fit in here.

JOB REQUIREMENTS

The existing models represent the daily cycle of attack/defense very well and can also be used to discuss security issues in general at a very high level. However, they do not work well for personal career growth. These models are based mostly around Doing and not Learning or Teaching. This makes sense, since most businesses are focused around getting things done. Getting things done directly affects productivity and profit, which is what matters to businesses.

Individuals, however, care about more. We care about doing better, and we also care about understanding better, making people's lives better, doing things differently, and doing different things. A straightforward model doesn't capture the inherent "squishiness" of human nature. We need something different.

So let's look at getting hired.

A detailed look at being hired is available in Josh More's book dedicated to this topic: *Job Reconnaissance: Using Hacking Skills to Win the Job Hunt Game* also published by Syngress. We will not repeat all that material, though you may wish to read it for yourself. As a summary, consider why an organization hires someone. In general, people are hired to solve a problem. In for-profit businesses, they are often hired because the product of their work can be sold for more than it costs to generate. In nonprofit organizations, people are often hired because their presence helps organizations achieve goals better and more cost effectively than without them.

With that in mind, consider two otherwise identical candidates. Both have no higher education, but one has a certification and the other does not. Odds are the one with certification will get the first offer. Now consider two new candidates. Both have college degrees, but one has taken the time to explore something and has written and published a detailed HOWTO document about it. Again, who do you think will get the job?

Fundamentally, proving yourself the best option in the job market is like proving yourself the best anywhere else. You have to stand out, in a positive way. Often, the smallest differences will make surprisingly large impacts. A single project can set you apart in a field of people that haven't bothered to do a project. In a slate of people without degrees, having a degree of any sort will matter. However, against a slate of people with degrees, a degree that matches your selected field is far more important. So:

Table 0.2.1

A person with...	Wins over a person with...
No degree, but with certifications	No degree
A general degree	No degree, but with certifications
A focused degree	A general degree
A focused degree and certifications	A focused degree
A personal recommendation from a mentor	A focused degree and certifications
An interesting project to discuss	A personal recommendation from a mentor
Experience in the "good old boys" club	An interesting project to discuss
An interesting project they have led	Experience in the "good old boys" club
An awesome life story	An interesting project they have led

Clearly, this is extremely subjective and based on the limited experience of the authors, with job experience in a specific region of the world. This model also loses the intricacies of different

types of degrees (associate, bachelor's, master's, doctorate) and ignores differences in people's interview skills. Instead we can model this as people accumulate points for each thing they've done. So:

Table 0.2.2

Experience	Points
Certification	1 certification = 5 points 2 certifications = 7 points 3+ certifications = 9 points
Degree	Associate = 20 points Bachelor's = 30 points Master's = 35–50 points, depending on job PhD = -10–100 points, depending on job
Personal recommendation from a mentor	Mentor not known to interviewer = 10 points Mentor known to interviewer = 30 points Mentor is the interviewer = 300 points
An interesting project to discuss	1 project = 20 points 2 projects = 30 points 3+ projects = 40 points
Experience in the "good old boys" club	Mentor not in the club = −10 to 10 points, depending Mentor in the club = 50 points
An interesting project they have led	1 project = 40 points 2 projects = 60 points 3+ projects = 80 points
An awesome life story	Story not pertinent to job = 20 points Story pertinent to job = 100–200 points

As you can see, having an awesome life story can trump pretty much everything here. Even if you may not be able to do the job as well as someone else, if those hiring you think you're awesome and want to work with you, you'll be considered the "best" for the job. If they think this, the hiring company will find some way to ignore requirements like degrees and certifications. You may have to get a degree or certification after you get the job, but you'll have broken in, and that's the goal. Some organizations will also support you in getting a degree or certification by paying for tuition, books, or exam fees.

Our job hiring model assumes there are common job types, and details those jobs, their requirements, and a rough path you may follow into the job you want. Ideally, you'll also create an awesome life story so maintain your story narrative as you move through your career. But having an awesome life is only half of it. The other half is that you must be able to tell your story.

This book will help you to find your new opportunities, maximize the outcome of each, and help you tell your story.

A NOTE ON EXPERIENCE — JOSH MORE

When I was young, being turned down for an ideal job because I lacked experience was both humiliating and angering. After all, I was smart and driven. I had the magic degree that was supposed to open doors. Running the whole "need experience to get experience" game was intensely frustrating.

Then, years went by and I found myself in the position of interviewing others. And I've got to tell you, the kids with degrees but no experience were arrogant idiots who had no idea how the real world worked. I hoped they'd get some experience because they'd be good after that, but I didn't want to be the one to break them in.

One way to get the experience you need to get a job that requires experience is to ignore the 40-hour cap on what you "should" be doing. *Should* is one of the biggest career killers there is. If all of your competitors get caught in the "should trap" and you avoid it, you automatically land in a position above them. For example, you could work 40 hours per week, and learn 10 more hours a week on your own time, and get a basic entry-level job. Then add projects to your resume so at the next career leap, you have more experience than everyone else who only put in 40 hours a week. This allows you to take advantage of demand for experience. If you're still in school, devote a specific time each week as well as time on school breaks towards developing the skills you need to give you a definite edge over everyone that doesn't. Identify your limitations and work on them until they no longer damage you, then focus where you can excel.

DEGREES

This book assumes that a degree is a bachelor's degree in the US collegiate system, with between three and five years of study with an emphasis in a specific field. The US also has associate degrees, which are usually two years of study, typically in more hands-on fields. US advanced degrees include master's degrees of usually two to four years, law degrees of two years, and doctorates which are usually six years or more.

The two-year associate degree typically fulfills a prerequisite toward a specific type of job, and are usually only offered by community and vocational colleges. Such degrees are useful when competing against high school graduates without training. Associate degrees also give you useful theoretical underpinnings for your intended field. Typical associate degrees that are useful in information security are in networking or system administration. Associate degrees are relatively cheap to get and can be attained fairly quickly. Degree programs for working adults on evenings and weekends are often for associate degrees. Beware of low quality associate degree programs and schools, don't trust the schools' own marketing, guarantees, or claims of successful graduates find your own contacts of current students and graduates, ask your mentor, and current and future employers about both good and bad schools.

Bachelor's degrees come in several flavors, but generally involve four years of study for a wider understanding for your subject matter as well as study outside of that area. A degree in physics, for example, will likely involve one or two classes each in math and physics for eight semesters. A degree in computer science would typically combine math and programming. There is also an expectation that you spend one to two classes each semester in unrelated disciplines such as literature, history, or psychology. The idea is that, by the end of your program, you have a more well-rounded understanding of the world and, along the way, have picked up the techniques you need to talk to people outside of your field.

Advanced degrees can involve between two and twelve years of study after your bachelor's degree, as you focus deeper on a specific issue in your selected field. Some very specific jobs require these degrees but, increasingly, they are seen as a liability outside of dedicated academic or research-focused fields.

No one can see the future. The more narrowly you focus, the more you risk getting it wrong. If you guess wrong with an associate degree, you're out two years of your life, and tuition. If you guess wrong with a PhD, you could lose a decade. However, if you guess right, you could land your perfect job and keep it for life. It's a high-risk, high-stakes game.

Many people reading this book will have a bachelor's degree (or equivalent) or be working on it. Such a degree will give you an edge over everyone who doesn't have one, but also doesn't require quite as much time or money as an advanced degree. The cost of advanced degrees is rising, and their relevance is being questioned more, so it is likely that we'll be seeing fewer of these in the information security market. Master's degrees in information security are uncommon and not well tested yet in the market. However, at the time of this writing, they are becoming more popular so unlike other advanced degrees, we expect we'll be seeing more of them.

A bachelor's degree tells a hiring manager that you've managed to stay focused and work within a system for four whole years. It says that you can get things done and won't cause trouble in the company. That's what matters to them. Many don't even care what field your degree is in. However, you can demonstrate these things without a degree once you've put in enough work elsewhere, but it may be more difficult to get that work and it may take longer than an associate or bachelor's degree.

Degrees typically only open the first few doors in your career. Not having a degree will make it harder to get some jobs, but when you have around 10 years, experience, you'll find that not having one will matter less and less to what you want to do. If you develop a pattern of doing awesome things and are known in relevant communities, lacking a degree may not matter at all.

CERTIFICATIONS

Certifications are sometimes viewed as a cheaper alternative to a degree, usually both in time and financial cost. They show that you've learned something and, if you work hard, that you can do something with what you've learned. Some certifications are more rigorous than others. The Offensive Security Certified Professional (OSCP) certification involves hands-on work attacking test systems to verify that you can actually perform a penetration test. Others, like the $(ISC)^2$ CISSP exam, are multiple-choice exams with required minimum years of information experience.

There are two challenges around certifications—which to get first and which to get last. A lot of people, once they get a certification, feel the need to acquire more. For more hiring managers, the first certification is much more important than the fourth certification. More certifications may add to your personal growth—and you may choose to let one certification expire and replace it with a more advanced one. However, generally speaking, more than three certifications are going to cost you more time and money than they'll add to your earning power. Certifications can also be perceived as negatives by experienced information security professionals who have personally experienced poor certification standards.

Part of the problem is that certifications are nearly always pass/fail for minimum requirements. Some feel that time and money spent on certification and yearly maintenance is better spent on practical work experience and study which go beyond any certification. It's not unusual for experienced information security professionals to allow their certifications to lapse. Some of them think so little of information security certifications that they don't approve of using certifications anywhere except as an otherwise useless but necessary evil to get through HR departments. They won't mention their certifications, omit certifications from their business cards, don't put them in job requirements if they can, and would prefer if others did the same. But this is a personal decision and the right choice is up to you.

Only a handful of certifications are considered requirements in the industry, and which ones change over time. These certifications are generally not the most respected by actual information security practitioners. This seeming paradox is created by the economics of certification. Human Resources (HR)

departments may require a particular certification, but HR departments don't know much outside of HR, especially in information technology and information security.

Large organizations like the US Department of Defense use certification as a standard of entry, and select one that their current employees in those roles can pass. So it must provide assurance, but also it must be possible to pass certification, otherwise it is not useful for those who accept certifications. The certification must also be well-enough known that the exam can be taken at any location that the organization has people. Ideally, it will also be vetted by a third party, such as the US ANSI or another standards organization. All of these requirements are expensive for the certification agency, so the certifications must also attract enough applicants that the certification fees cover the cost of the program. All of these requirements result in a market environment in which the best-known certifications are weakened (to increase passing rates). Some of the inexpensive certifications are extremely easy to pass, as they have to increase the number of applicants to cover the cost of managing the program. Thus, less-well-known, but more highly respected certifications are created in response to the belief that certifications "aren't what they used to be," as people want their certification to be something they can be proud of.

So how do you choose a certification? In our experience, your first certification should directly apply to your new job. A good listing of door-opening certifications can be seen on the US Department of Defense's Approved 8570 Baseline for different roles reproduced below with roles **in bold**:

Table 0.2.3

IAT Level I A + -CE Network+ CE **SSCP**	**IAT Level II** GSEC Security+ CE SSCP CCNA-Security	**IAT Level III** CISA CISSP (or associate) GCIH GCED CASP		
IAM Level I CAP GSLC Security+ CE	**IAM Level II** CAP GSLC CISM CASP CISSP (or associate)	**IAM Level III** GSLC CISM CISSP (or associate)		
IASAE I CISSP (or associate) CASP CSSLP	**IASAE II** CISSP (or associate) CASP CSSLP	**IASAE III** CISSP - ISSEP CISSP - ISSAP		
CNDSP Analyst GCIA CEH GCIH	**CNDSP Infrastructure Support** SSCP CEH	**CNDSP Incident Responder** GCIH CSIH CEH GCFA	**CNDSP Auditor** CISA GSNA CEH	**CNDSP Manager** CISSP-ISSMP CISM

As you can see, $(ISC)^2$ CISSP, EC CEH, CompTIA Security+ CE, and SANS GCIH are fairly popular certifications across the board, so these might be good choices for your first certification. They're well-known and likely won't work against you. They are not, however, technically equivalent. If you go to the specific certification outlines and training agendas, you'll see that they cover entirely different things. See the Appendix: Certifications for URLs.

But, if these certifications are equivalent from a hiring perspective, it makes sense to pick the one that will teach you the most and cost the least time and money. A good method of selecting which certification is best is to look at how much you'll learn. The more learning you do for a certificate the more valuable it will be over the long term, not because the certification itself has that much intrinsic value, but because the certification learning process will be with you for the rest of your life.

As an example to measure the personal value of a certification, compare the SANS GCIH to the EC CEH. In the two tables below, the certification objectives are placed one-per-line and a subjective guess is made as to how much new material would be learned in that area. For example, a developer reviewing the list might already have detailed understanding as to what a buffer overflow is and score it low at 10%. However, a network administrator might not know much about buffer overflows and score it high at 80%.

Once both certification lists have been scored, average the scores for how much new learning would be involved with each certification. In the examples below, we are assuming a skilled but moderately inexperienced person is comparing the two certifications.

Table 0.2.4

SANS GCIH New Learning for Skilled but Moderately Inexperienced	
	% New Material
Backdoors & Trojan Horses	80
Buffer Overflows	80
Covering Tracks: Networks	80
Covering Tracks: Systems	80
Denial of Service Attacks	10
Exploiting Systems using Netcat	50
Format String Attacks	80
Incident Handling Overview and Preparation	80
Incident Handling Phase 2 Identification	70
Incident Handling Phase 3 Containment	50
Incident Handling: Recovering and Improving Capabilities	50
IP Address Spoofing	80
Network Sniffing	10
Password Attacks	40
Reconnaissance	50
Rootkits	30
Scanning: Host Discovery	20
Scanning: Network and Application Vulnerability scanning and tools	40
Scanning: Network Devices (Firewall rules determination, fragmentation, and IDS/IPS evasion)	40
Scanning: Service Discovery	40
Session Hijacking, Tools and Defenses	80
Types of Incidents	70
Virtual Machine Attacks	90
Web Application Attacks	80
Worms, Bots & Bot-Nets	60
Average new material:	**57.6**

Table 0.2.5

EC CEH Learning for Skilled but Moderately Inexperienced	
	% New Material
Introduction to Ethical Hacking	10
Footprinting and Reconnaissance	50
Scanning Networks	20
Enumeration	50
System Hacking	80
Trojans and Backdoors	60
Viruses and Worms	60
Sniffers	10
Social Engineering	50
Denial of Service	10
Session Hijacking	80
Hijacking Webservers	80
Hijacking Web Applications	80
SQL Injection	70
Hacking Wireless Networks	70
Evading IDS, Firewalls and Honeypots	80
Buffer Overflow	80
Cryptography	40
Penetration Testing	50
Average new material:	**54.2**

For this person, the SANS GCIH certification at 57.6% new material is a somewhat better choice from a learning perspective compared to the EC CEH at 54.2% new material. But this is just an example. A simpler approach would just count the number of new enough items. What is "new enough" is also subjective, but using 60% or less has similar results: SANS GCIH is 13 and EC CEH is 11. Note that these numbers are just examples and would always be subjective and dependent upon each person.

A deeper but still subjective approach would review a complete study guide for each certification and measure new material by the number or percentage of pages, paragraphs or lines of new material. A short cut would only look for new material in the glossary or index. If practice certification exams are available for free or cheap, then take them, and whichever score is lowest determines the certification to pursue. Again, the goal is to learn, not to accumulate certifications.

If money is limited and you have to pay for it yourself, it may be wise to compare the certifications in terms of total dollars. Include exam preparation costs, the exam cost itself, any travel or time off needed to take the exam, and certification maintenance costs. Some certifications have effective

maintenance costs of hundreds of US dollars a year. Some exams are only infrequently offered in some cities requiring possibly expensive travel. Others have reduced costs for retaking after a failing score.

There are three ways to pursue certification: class, self-study, or directly challenging the exam.

Table 0.2.6

SANS GCIH vs EC CEH Costs				
	Price	**Percent New Learning**	**Cost of Knowledge**	**Wasted Money**
EC CEH—Exam Only	$600	54.2%	$325.20	$274.80
EC CEH—Courseware + Exam	$825 + $600	54.2%	$772.35	$652.65
EC CEH—Class + Exam	$2,895 + $600	54.2%	$1,894.29	$1,600.71
SANS GCIH—Exam Only	$600	57.6%	$345.60	$254.40
SANS GCIH—Courseware + Exam	Not Available	Not Available	Not Available	Not Available
SANS GCIH—Class + Exam	$5,095 + $600	57.6%	$3,280.32	$2,414.68

Here, by calculating the amount of the money you're spending that goes only towards new knowledge (percent times price), you can determine cost of knowledge. You can then subtract this number from the total price to determine whether that is the best use for such money. The wasted money—money spent to learn things you likely already know—is the measure of how valuable the certification is to you. Clearly, if the person in this example thinks they can study on their own without purchasing the courseware, taking the GCIH in the Exam Only mode is the best way to go. However, if they need a class, perhaps the CEH class is a preferable option, given that it wastes almost $800 less than the GCIH option.

How these numbers break down will vary drastically based on the specific certifications you are comparing and your specific skill level in each. In some cases, such as defending a job you already have, you might already know the majority of what you'll be tested on, so new knowledge is low. If the price is low enough, it may be worth doing just to keep your job, even though most of it would fall under "wasted money," If, however, you want a challenge, it may be worth it to pursue a much more highly priced advanced certification, because the wasted money count is low, so most of your spending will go toward new learning.

In general, many people get their first certification in a way that is as easy as possible, simply to get that edge over otherwise comparable people. Subsequent certifications tend to be far more challenging; the direct financial effect is minimal compared to the joy of learning and the increased effectiveness gained.

STRIKING A BALANCE

In the end, you need to strike a balance. By considering where you actually are, you can decide how much effort it is worth to try to do more. This may involve investing time in your education, projects to boost your experience, or studying for certifications. As you get older, you will likely find more of your

time going into interesting projects rather than education and certification, which provide diminishing returns for experienced professionals.

However, you will likely go through a period in your life where it's tempting to go for more and more schooling. After all, most of us spend 16 to 20 years of our lives in school. That's the safe option. If you graduate and don't immediately find a job that you "deserve" and are tempted to go to grad school or pursue another degree, consider whether you actually need it, or whether you're just discovering that the real world is harder than you were led to believe. Many information security professionals have considered advanced degrees. However, as shown in some of the stories included throughout this book, many information security professionals have found successful, useful, and rewarding careers without them. Sometimes you just have to roll up your sleeves and get to work.

It won't be easy. There will be pain and frustration. That's part of life.

However, information security has more pain and frustration than some other fields. Your organization's adversaries are well-funded by criminals and/or nation-states. Your bosses and customers don't understand the issues, and you will not have enough time and money to build the solutions you want. However, if you can get past this—something we've learned by experience—you can find yourself making real differences in people's lives and getting paid well to do it.

For us, it's worth it. If you think that applies to you, read on.

CHAPTER

MODEL FAILURES 0.3

BARRIERS

> "The map is not the territory"
>
> — **Alfred Korzybski**

Models are imperfect representations of reality. Problems can arise when you measure your progress against the abstract perfection that's inherent in the model you are using. This book uses a generic approach to gaining or improving employment within the information security industry. It will work in some environments, but will fail (sometimes spectacularly) in others. In general, though, working from a solid model will help you to identify and overcome barriers you may experience in the working world.

You can deal with barriers to the job you want by pushing through them or by subverting them within the system. Different barriers require different approaches.

BARRIER ENERGY

The first barrier you are likely to encounter is that of energy. Getting a new job isn't easy, especially in a field new to you. It's hard to work for an uncertain outcome. It's possible to expend too much effort for too little reward. Progress isn't straightforward, so each thing you do gives you a new or better tool to use in working toward your goal, and perhaps gets you one step closer. Success is not guaranteed. You're building skills, not checking off items on a guaranteed plan. There *are* no guaranteed plans; failure itself is learning, so gain energy from learning, and don't lose it from failure. Once you've started, it's often easier to keep going.

In the field of chemistry, there is a concept called "activation energy." This concept holds that some chemical reactions require a certain amount of starting energy to happen. A good example is that of combustion. A puddle of pure alcohol will just sit there evaporating to nothing. However, if you apply a flame, something rather different will happen—it will ignite. The same is true, metaphorically speaking, with doing the outside work needed to get a new job. Many people have trouble just sitting down and doing what they feel needs to be done. Often, the hardest tasks are shifted to the bottom of the "to do" list. Thus, you may find yourself organizing your digital music collection three times to avoid cleaning out the garage, or cleaning out the garage instead of looking for a new job. With learning tasks, a "hard" task tends to be one that involves more learning than others.

You may find that learning a brand-new skill is much harder than improving those you already have. Improving current skills is great if you want to move further up in your current job. It's not as good if you want to transfer into a different field. If you find yourself constantly putting off tasks

because you're tired or because of something else that must be done, you may need to apply a large amount of activation energy to get yourself started or you need a catalyst. For example, as we worked on this book, in fact, there were some tasks that were just too hard for us to do individually. Instead, we set aside specific "sprint" days, where we'd hold one another accountable and push through the barriers blocking us.

A common task catalyst is the dedicated work day and dedicated work area. If you can afford the time, reserve four to eight hours and plan to do nothing but focus on your required task during that time. A dedicated work area and equipment is often helpful and sometimes required for security work. Don't include work area setup or tear down in that time period. Instead, do it ahead of time so you can immediately start work when the clock starts.

Minimize distractions:

Block out your schedule ahead of time and let people know so they won't interrupt you during dedicated time.

If your work space has a door, close it, if there's a window with distractions, block it.

Silence your phone, alarm clocks, and any other devices, including turning off vibrate. It may be easiest to turn them off, put them into airplane mode, or put them out of sight but where you'll easily find them again. There are apps for that.

If it's noisy, use ordinary, cheap, readily available ear plugs. Don't use ordinary ear phones which if turned to high volume to mask noise will also damage your hearing. Noise canceling ear phones help, but cannot block all noise, especially voices, will cost more, and are yet another technological distraction.

If you need to listen while working consider instead *noise isolating* earbuds, also known as canal phones or in-ear monitors (IEM), which seal inside the ear canal and passively block outside noise and sound. The cost is comparable to good ear phones but with much better isolation, especially with custom fitting by a professional audiologist.

Block visual distractions by running applications in full screen mode, especially text editors and word processors. Turn off task bars and notification areas. If you have multiple displays, but don't need them, turn them off.

For deep concentration avoid any background noise because any sound, even soothing music, is distracting.

If you don't need the Internet to accomplish your task, turn it off. If you think you do need the Internet, seriously reconsider and instead make a list of what you need and do it later. Defer your procrastination. If you find yourself constantly going to different social media sites, consider using web filtering to temporarily block those sites, or unplug and disconnect during this time. The goal is to focus and take less time to get into the deep concentration need for thoughtful work. If you are trying to light a puddle of alcohol on fire, a flame applied for a few seconds is more effective than shining a heat lamp for several hours. If you have an energy problem, focus the energy you do have as much as you can and you're much more likely to solve it.

This focusing limited energy is like borrowing from the future. You may find that the technique works well, so you do it over and over again, only to find that eventually it stops working for you. The problem is that you can't create energy from nothing. When you think you're being productive by cutting out all distractions, you are also likely cutting out those activities that allow you to rest and regain energy for the next round. If you start getting sick or extremely tired, you may just need to take some time off from the focused tasks to recover. If you don't, you risk burning out completely and losing focus for months.

However, most of the barriers you'll encounter will be external to you. These cannot be addressed by simply buckling down and pushing your way through. Most are surmountable, but you have to question whether they are worth fighting. What follows is an incomplete list of barriers that you may encounter on your way to your information security dream job.

HUMAN RESOURCES

Many organizations have a Human Resources (HR) department. HR is supposed to keep the organization out of legal trouble, handle interpersonal conflicts, and, to varying degrees, manage benefits. Many such departments also become involved in the hiring process. This typically happens because there are a lot of time-consuming steps involved in hiring someone, and it makes economic sense to move those tasks off the plates of the hiring managers, so they can manage people. However, it often results in a situation in which you have people unfamiliar with the actual job performing the initial filter for candidates. This trend within business is largely responsible for the "you need experience to get experience" trap.

Dealing with this barrier is particularly difficult if you are trying to move into a job you've never done before. HR personnel tend to follow the old "no one ever got fired for buying IBM or Microsoft" approach to decision making. If you look like a risk, you'll be passed over for the safer choice. Thus, you have two options. You can repackage yourself to look safer. This involves spending more time developing a work history in related fields, probably by doing additional work on a volunteer basis. However, the second option is more likely to be successful— bypass them altogether.

By taking the bypass approach, you find a way to meet the hiring managers directly and get far enough along in the process that by the time you meet HR, you've already been approved, avoiding the whole appraisal process. There are many ways to do this, most of which are detailed in Josh More's book *Job Reconnaissance*, also published by Syngress.

OLD BOYS' CLUB, RACISM, AND SEXISM

The phrase "old boys' club" refers to the tendency of people to want to work with people just like them. It brings up images of old white men sitting in private clubs smoking cigars and giving favors to their friends. However, the attitude itself is universal. Fundamentally, most people want to work with those who are like themselves or those they have worked with before, regardless of other differences. It's more comfortable. However, evidence strongly suggests that more diverse workplaces are more effective, more successful and more profitable. Pointing this out, of course, may get you kicked out of the club.

Whether the tendency to prefer their own type applies to previous employment, race, sex, or class, the first decision is always going to be the same. Will you be comfortable in a job where you have to either try to blend in or constantly fight the status quo? It's okay to not want to do either. That means that the organization isn't a good fit for you and that you're not a good fit for them. Find one of their competitors, add diversity there, and out compete the other firm.

If, however, you decide you want to try to blend in, learn the cultural markers. This is easier to do in some cases than others. In the United States, an individual who appears to be white, male, and reasonably (but not overly) educated will have a lot more options than someone who presents otherwise.

If you're comfortable hiding aspects of who you are from your co-workers, at least initially, that may make it easier to find a job. People with different political opinions, belief systems, sexual orientations, and gender expressions from the majority can often find reasonable employment and later, as they prove themselves an asset to the organization, slowly let more of their personality out. The cost to this approach is possibly never being fully comfortable in that workplace. It should also be noted that some people will simply not be able to blend in to all environments. Some environments won't be compatible with all people, you can learn how to speak and act in different ways and become familiar enough with specific interests to blend in.

There are increasingly more jobs where little to no physical presence is required, sometimes called full-time telecommuting or 100% telecommuting. This is common enough in the open source communities and some organizations that are conferences just so people can physically meet. These events are usually not mandatory, with most work still done remotely and over the Internet.

If you are a natural fighter, odds are that you've been doing it for a very long time. You probably had to fight your way through school and any previous job experience. It is unlikely we can give you any advice on how to fight that isn't insulting. The techniques you use to combat racism and sexism in the workplace will be uniquely personal and vary with your environment. So, instead of tips on how to fight, we instead suggest that you consider these things:

1. Security is about protecting others. If an organization recognizes and values that, you can find common ground in common protection. If it does not, the job will be undervalued. If you are personally undervalued because of irrelevant personal details, and are doing an undervalued job, think strongly on why you want to do it. It may be a great stepping stone, but it is very unlikely to ever become the job of your dreams.
2. Business is about making money. Most people in a high position in business recognize this, and the belief in the almighty dollar can trump other prejudices. As with protection, if the company can make more money with you than they can make without you, you have your "in" into the business.
3. An organization's mission is about accomplishing goals. Sadly, nonprofit organizations can be some of the most racist, sexist, and otherwise badly functioning organizations out there, largely because there is no counterbalancing profit motive. Nonprofits can also be some of the best places to work, for the same reason. If you can show that an organization can accomplish its goals better with you than without you, you better overcome these barriers.

These three points can get other people to fight for you. In a hiring situation, there may be a group of people influencing the decision, but usually only one person making it. If that person is on your side, they can fight for you. If they're not, but enough people are on your side, you may find yourself with a job offer anyway. This indirect fighting is far more likely to succeed than any sort of direct fighting.

Direct fights, such as involving the law, are expensive and less likely to work. Even if you do succeed with such a tactic, you'll always be known as the person who sued your way into the job. Once you get the job, continue the fight to keep it, advance, and to change the culture. Join organizations and community groups such as those listed in the Appendix, and discuss ways to improve acceptance and increase diversity.

Remember, at this phase of your professional development, your primary focus is on overcoming the barrier to entry, not fighting the constant uphill battle for universal equality. That fight can wait until you're in the door.

CORPORATE CULTURE

An organization's corporate culture is something you may never understand fully, and almost certainly will not understand before working there. There are many books on corporate culture that may be worth reading, especially those for a particular company where you wish to work. Instead, we want to explore how corporate culture can prevent a job offer. Some companies have unique, unusual cultures that may be difficult to understand. Particularly notable organizations will have articles and books written about them; look for them and read them as part of your job reconnaissance. Current employees may be able to help, but ex-employees, part-time employees, and contractors who have experience outside the organization may be better able to explain the culture.

Sports Culture

In an organization with a sports culture, most employees will have played a sport in school or college. Sports metaphors will be used in most meetings. Everyone will have a favorite team and, most times, it will be the same local professional or college team. There will be an overall belief that, when asked, you should "be a team player" and "take one for the team." You may be expected to work late nights or weekends or even be publicly called out in a meeting so the business can save an important client relationship.

If you are a person who would fit into such a culture, none of this sounds bad. If, however, you were never one of the "jocks" or "athletic supporters" (somewhat likely if you're reading a book on information security careers), it may sound awful. However, if you still want to work in such an organization, you have to get in the door. To overcome this barrier, check out the Facebook and Twitter accounts of the people who work there. Figure out what sports they talk about and which sports teams are involved. Pick one team and learn the names of the important players. Go to YouTube and watch the highlights of some games. Learn the rules for the game and review the last few years of results. Don't lie, don't pose, don't pretend to be a fan—the goal is be inoffensive, culturally sensitive, informed, and not give a blank stare when they make some sports reference. When asked about it, answer truthfully, such as "I don't really follow sports much, I've just seen some <team> games." This approach can help you blend in without lying or otherwise being dishonest.

This same approach can work for other corporate cultural interests, see below. You might even find a new interest yourself.

Education

Most organizations have one or more base education levels. Many small businesses expect everyone to have a college degree. Some startups involve everyone having a master's degree or even a PhD, while others are made up of current college students or college drop-outs. Many large companies expect employees to have at least a high school degree. These are, however, just basic expectations and not every organization will fit the pattern. What will fit, however, is the overall expectation that you be able to communicate on the same level as the rest of the team. Generally speaking, the more educated a person is, the more abstractly they speak.

Cultures centered around people with less education tend to be more focused on day-to-day issues and less on the long-term goals. If you don't fit such an organization's model and still want to work there, you need to explain how you will make their day-to-day work easier. Find out what they don't like to do and make it clear that you don't mind doing that and perhaps you can improve the process. Cultures focused around greater levels of education will tend to speak more strategically. They want

to achieve big dreams and by focusing on that you can blend in even if you don't have the expected degrees.

An exception to these "high" and "low" education-based organizations is that of artistic organizations which tend to be far more accepting and less focused on general education. The artistic part of an organization may be most of the company, or a sub-group such as marketing, or information technology. Some of these organizations have a high number of people interested in art, books, and film. Some interests are general, others are highly specific. Within information technology groups, science fiction fandom and computer gaming are common. However, if you—for example—mistake Star Wars for Star Trek, the response can be brutal. These organizations follow a basic educational path except that you are expected to be highly educated in the specific form of art that is preferred by the group.

Like a sports culture, you need not be an expert; just familiarize yourself enough with the core interest to be able to talk about it reasonably and also know when to admit your ignorance. If you need cultural references, Wikipedia is excellent for nearly all popular areas, even sports.

Military

Organizations made up of mostly ex-military personnel exist within the information security community. These tend to come in two types.

The first type consists of people who think the military had it right and are trying to run their organization like a military unit. These are the no nonsense highly driven people you often see depicted in war movies. You may think they're just a stereotype, but they do exist and some people thrive in that culture. If you have not served in the military, you will likely not fit in, regardless of how hard you try. Having served in a different military service or different country's military may also be a problem—not only is the culture going to be at least somewhat different, but it may even be antagonistic even if from the same country. For example the rivalry between the US Army and the US Navy is notorious.

The second type have a militaristic preference, but employ a number of those who have never served in the military. These organizations share a common background that is stronger than that of education or sports, but are more welcoming. You may encounter some terse speech and intolerance for mistakes, and a militaristic chain of command, but in general, these organizations are very goal-oriented and can be great places to work. If you wish to work for one of these, it really helps to appreciate the work that the military has done. If you have negative opinions about recent military involvements, it is generally best not to express them; you hear them expressed by someone who has actually served. Even then, be careful to keep criticisms to those who sent the military to war and not to those who actually fought. Militaries are not democracies, and neither enlistees or draftees get much choice in how they serve.

Academia and Health Care

Academia is, oddly, both extremely similar to and the exact opposite of militaristic cultures. Academic institutions are structured such that the most educated people, professors and doctors, are at the top and everything done by the organization exists to support them. This structure exists in healthcare, universities, and other organizations started by those doctors and professors. They are every bit as idealistic and driven as the military-based cultures, but lack a team focus. In academia, it is very much "everyone for themselves." Sure, you can make great friendships, but don't expect anyone to put themselves in figurative harm's way to help you. This is very different from a military organization, where putting yourself in literal harm's way for your team is part of the training and often part of the experience. Fortunately there is much less physical harm in academia, although the fighting may not be any less fierce.

To break into an academic culture, be prepared to defend yourself intellectually. Any time you need, not just want, to point out that someone is wrong, have the proof and be willing to enter a debate, using the scientific method where possible, because your opponent will also. If you can play the game without making those in power look bad or directly telling them no, you're in. Note that intellectualism and scientific method by name or by practice will not work in some other environments. It may not be understood, or worse, understood to be academic and consequently derided.

AGE

Ageism isn't talked about nearly as much as sexism or racism, but as the world's population gets older, the issues are growing. Older people tend to be more expensive employees because of their accumulated salary and compensation requirements and increased experience. This isn't exactly ageism but if a company can hire someone for half the cost of someone else, and believes them to be otherwise equivalent, of course they're going to go with the less expensive candidate, who will probably be younger and have less experience. The trick here is to make it clear that you're better than the other candidates (or be willing to work for less). Use the techniques discussed in Chapter 4.0 "Boosting" of this book, also see the branding suggestions in Josh More's book *Job Reconnaissance*.

Ageism can also work the other way: younger people may be wrongly assumed to lack the skills or experience to do the job simply because they are younger. Such assumptions are as ageist as assuming older people can't learn new skills or know new technologies.

To combat straight-up ageism, you must demonstrate that you can do the job. Information security isn't like working in a warehouse. If your brain works, you can do the work, even if your body is older. If you leverage your skills and experience and show that you can out think your competition, you should be able to get through the door. Some people report successfully using blending techniques as mentioned above.

Page left intentionally blank

CHAPTER

TIER 1—LEARN

1.0

LEARN/DO/TEACH

> "In theory there is no difference between theory and practice. In practice there is."
>
> **— Jan L. A. van de Snepscheut**

The key to this book and, we believe, to life in general, is the concept of Learn/Do/Teach. This concept was first developed in the medical field, where medical students first learn a medical procedure, then do the same procedure while guided by someone experienced, then teach a less experienced student the same procedure. Experiencing a concept directly multiple times from multiple perspectives results in better understanding and retention.

Unless you are extremely unusual, the first time you learn something, you are unlikely to learn it in any great detail. You may understand it roughly, but it is unlikely that you understand it well enough to have any level of mastery. Once you start doing something with it, however, you can rapidly find where your understanding fails. When theory meets reality, reality always wins, and it will be common for you to discover where the initial theory omitted some details. The more time you spend in Do, the greater you refine your understanding. Finally, in the Teach phase of learning, you get someone else through the Learn step. This further enhances your understanding, as you find areas where other people's understanding comes into conflict with yours, and you resolve the conflict by better understanding them and the material.

WHY LEARNING MATTERS

Learning is critical in a career. At one time, perhaps, people could do the same job every day for a lifetime and not have to learn anything new. However, so long as there is someone or something that will do the same work for cheaper, you are forced to improve, or learn a completely new job. Over time and at quantity, this tendency toward continual improvement sets person against person, company against company, and nation against nation. Fundamentally, the best Learners (as they become Doers, then Teachers) drive the economy ahead.

Traditional jobs are going away. It is important, however, to understand what "going away" means. The Earth is a closed system. We can't lose a significant number of jobs any more than we can lose a significant amount of water. While a small amount of either may vanish over time, this amount is

negligible. Instead, when people speak of "lost jobs," what they mean is "structural unemployment" where society either perceives less value for that particular work or there is a less costly way to achieve the same work. In the former, odds are that something has replaced it and things that used to be valuable simply no longer are. If jobs are "lost" during this process, they invariably reappear somewhere else in the economy.

For example, aluminum is now used for disposable, single-use beverage cans used to be a precious metal. As an example of conspicuous consumption, Napoleon III reserved a prized set of aluminum cutlery for special guests at banquets, while less-favored guests used gold knives and forks. However, cheap industrial scale bauxite ore refining, and mass production, made and anything made from it cheap. The latter case, however, is often tied to competition.

None of this means that in the case of a job shift, skills will be transferable. Quite to the contrary. In the same way that Earth is a closed system and species that cannot compete die out, skills that are no longer needed, due to improvements in technology or changes in what the market demands, will similarly die out. The trick is to constantly be adding new skills so when this, inevitably, happens, you are not effected as much as others.

This is a book about information security. It will, by necessity, occasionally drift into the philosophical, but fundamentally, we assume that you are reading this because you want to get into the field. Within information security, the good news is that, unlike elevator operators, ballast heavers, and scutchers, most jobs in information security are unlikely to go away because of technological advancement. However, specific jobs tied to particular vendors or very specific technology are subject to the fashions of the industry and organizations. An elevator operator who could only operate an Otis and not a Schindler or General Electric would have had a difficult time. The trick is to aim toward the general, while still being specific enough to be of value to prospective employers. The more specific you are, the more value you have to an employer. However, the trade-off is that such a career is limited. As technology advances, common tasks get automated, both in terms of attack and defense. So from a strategic perspective, it is better to focus on specific threats and automate their defense than to focus on an entire class of attacks that could eventually be managed by a standalone appliance.

Attackers and defenders are in an arms race. An attacker finds a vulnerability and creates an attack to exploit it. A defender may detect this attack, and if the defender is not completely destroyed, then recovers and implements a countermeasure, so the attacker is driven towards a new attack or even counter-countermeasure—repeating and moving ever faster with each cycle. This cycle affects all attackers and defenders and results in an ever-changing environment where advantages change moment to moment.

WHY LEARNING MATTERS TO YOU

Attaching yourself to a specific technology or even a specific area of compliance is short-sighted. If your chosen technology is proven to be ineffective, you may find yourself at a dead end—both unable to find more work in your field and unable to find time or money to learn something new.

This is why learning is critical to you. Only by constantly learning can you keep in front of the wave as old technologies drown in the marketplace and sink into disuse. Also, only through learning can you determine what new areas might exist in the future and position yourself to leap onto the new technology platform as your old one dies.

You can make a good living with older technologies. However, as those technologies become common, it results in job market crowding, which can drive down salaries and even result in unemployment due to too many people being available for a specific job.

By focusing less on specific technologies and more on learning, you become nimble, leaping from platform to platform as the market changes.

WHY LEARNING MATTERS TO COMPANIES

Learning is also critical to companies. Since the attackers are constantly learning and creating a new deluge of attacks, defenders must respond. An organization must decide whether to learn or to hire. When one organization learns and uses that knowledge to protect others, it basically functions as a product or service. However, products are easily analyzed and attackers put a disproportionate amount of effort into finding flaws. After all, if one product can be shown to be flawed, all organizations protected by that product are vulnerable.

A company is in the same situation you are. By connecting itself directly to a specific product, it faces the risk of that product eroding over time and leaving it vulnerable. However, by not doing that, it has less time to devote to targeting its customer base, resulting in lower profit margins.

By focusing more on learning, companies can take greater advantage of new technologies, either to improve efficiencies (costs) or to gain advantage (value).

HOW TO LEARN

There are many ways to learn within information security. As is common in technical fields, there are many who function well as self-learners. These people learn best from books, videos, or tech articles. Others learn better when being guided, either by taking classes or under direct mentoring of another. There are learning opportunities both online and offline. The key, once you identify what you need to learn, is to identify how you best learn and maximize your use of time.

Identifying what you need to learn is what the rest of this book is about. How to learn how *you* need to learn, however, is addressed here.

The first step is to identify how much you need to learn before it can be called "done." Few people are used to thinking this way, as school was highly structured and much learning after that worked under a model of "poke at things until they work, then stop." This is a perfectly functional model for learning about many things, but it doesn't serve you well when it comes to security. Security professionals talk a lot about the idea that people either have a security mindset or they don't—that security thinking is somehow inherent to a person. This is not necessarily the case.

In truth, thinking about security involves not just understanding how things are built, but also how they break, how they can be broken by people deliberately, and what other people would gain by breaking things. Thus, one way to measure how much learning is necessary is when you can answer the following questions:

- Do I understand why it works the way it does?
- Do I understand at least three ways it can fail?
- Can I list at least three ways someone else can benefit if it fails?

One approach to learning is to keep at a subject until these questions are answered. That gives you a minimum bound for learning. However, some things are harder to learn than others, so it may be helpful to place a maximum bound as well. First, decide the worth of what you learn in hours of your time. If it's only worth one hour then set a timer or alarm clock for one hour ahead. If you can answer the above questions in an hour, great. If you can't, then stop after an hour, since you already decided that it's not worth the cost to learn it. Don't fall into the trap of sunk costs, where even more is spent because so much was already spent.

MENTORS

A mentor is someone who can suggest, guide, and advise. There are many kinds of mentors, and a person can have more than one mentor. Even mentors will have mentors. A career mentor helps with the career path, such as what skills to develop, and what certifications, if any, to pursue. A skills mentor helps with the particular skills. Technical skills mentoring is important, but social/soft skills mentoring can be even more important, especially for the technically minded. Good social skills and soft skills will elevate the technically minded far above those who lack these skills.

Corporate and industry culture mentors can be very different. Corporate culture can vary considerably. Industries also have cultures which will be common in companies in that industry. Mentors can help with navigating the difficult aspects of these cultures. This means that a mentor from a different company but in the same industry can as useful, or even more than a mentor in the same company. The outside mentor can provide the perspective that all companies in the industry are alike in some way, or not, as it may happen.

A particularly experienced mentor may also provide direct longer-term instruction as part of a master/apprenticeship. Although the master/apprentice terms are somewhat archaic, the old traditional concept is coming back in some industries as an alternative to both formal, structured education and informal, unstructured on-the-job training. Traditional apprenticeships consisted of on-the-job-training under a master and would last for several years. At the end of the training the apprentice would become a journeyman who would typically leave the master and set up their own business. An all-too-common tradition was abuse of the apprentice by the master — having the apprentice be involved in work, but not involved in learning new skills, thus protecting the master's livelihood. Although modern-day formal apprenticeships are rare, the shorter term unpaid and often exploitative internships are common, should the opportunity arise, prospective employees should be aware of their rights.

Finding a Mentor

Some professional organizations and some companies have formal mentoring programs. If your organization or company doesn't have a mentoring program, consider starting one. Management support is particularly helpful here, but informal mentoring can still work and may be a natural extension of a work-related relationship. Mentoring programs may also be available through local young professional groups and trade associations.

A mentor needn't be in the same company or even the same industry. Local clubs, meetups, social networks, and other gatherings can be a place to meet a mentor. See the Appendix: Community for more details. Once found, identify how you will work together and what each of you will get out of the relationship. Some people work best with a face-to-face meeting each month. Others can work fine via email, telephone, or

social media. In identifying the purpose of the mentoring, try to be more specific than "I want to learn from you." Set goals, such as a 20% salary increase within two years, achieving a specific certification or job title, or even just laying out a general road map to get you where you want to go. Everyone works better with goals.

Being a Mentor

Mentors are similar but not identical to teachers or tutors; see the Teach chapter for more information on being mentor.

CLASSES

Classes work in a similar way as mentorship, but money has to be factored in as well as time. When you take a class, you must expend both time and money. Perhaps you have an employer willing to pay and you can discount the money aspect somewhat, but it is still going to take time. If you really want to break into a new job, you have to consider how much learning you can get for your time and financial resources.

One way to do this is to assign an estimate for the value of learning. Learning something new could result in a new job at more pay and greater happiness, or just reduce the stress you find in your current job. Before you invest in a class, consider first if there is any other way to learn what you need to learn. Then, figure out how well you will learn that way. Only then can you make a decision as to whether or not that approach makes sense.

Fortunately, legitimate classes provide an agenda or syllabus. Much in the same way we measured learning for certifications, we can measure classes for learning. As merely an example, let's look at an old but representative version of SEC401 from the SANS Institute. This is one of the classic starter classes for getting into information security via a training/certification approach. The syllabus for day 4 is listed below:

SEC401 Syllabus—Day 4	
Network fundamentals • Network types (LANs, WANs) • Network topologies • Ethernet, token ring • ATM, ISDN, X.25 • Wiring • Network devices • Voice Over IP (VOIP) IP concepts • Packets and addresses • IP service ports • IP protocols • TCP • UDP • ICMP • DNS • IP behavior • TCPdump • Recognizing and understanding	• UDP • ICMP • UDP Behavior IOS and router filters • Routers • IOS • Routing • Routing protocols • Access control lists Physical security • Facility requirements • Technical controls • Environmental issues • Personal safety • Physical security threats • Elements of physical security

(Continued)

Cryptography • Need for cryptography • Types of encryption • Symmetric • Asymmetric • Hash • Ciphers • Digital substitution • Algorithms • Real-world cryptosystems • Crypto attacks • VPNs • Types of remote access • PKI • Digital certificates • Key escrow • Steganography • Types • Applications • Detection	PGP • Installing and using PGP • Signing data and what it means • Key management • Key servers Wireless • Common protocols • Common topologies • Misconceptions • Security issues • Securing wireless Operations security • Legal requirements • Administrative management • Individual accountability • Need to know • Privileged operations • Control types • Operation controls • Reporting

To assess the value of this class to you, score each item by how well you already know it and how you perceive its market value. Create a score list as seen below. Assume that the maximum any skill may be worth is $100 and the minimum is $0. The "Learning" column is a percentage of what's left to learn. For example, if you feel that you know 80% of VoIP, then you have 20% left to learn. This approach may not be completely fair but, as a rule of thumb, if you have no idea what's left to learn, just score a column as half value, such as $50 or 50% learning and be done. But if you don't know what a term means, assume $100 or 100% learning value in that item.

Learning Objective	Value	Learning	Learning Objective	Value	Learning
Network types and topologies			Symmetric Cryptography		
VOIP			Asymmetric Cryptography		
TCP, UDP, ICMP			Hashing		
DNS			VPNs		
TCPdump			PKI		
Routing Protocols			Steganography		
Physical Security			PGP		
Technical Controls			Wireless		
			Legal Issues		

For illustrative purposes, let's assume two different people are going through the exercise. Azal has been working in technology for a while, specifically in the networking area. Day 1 is likely to be review for him. Morgaine is just graduating college with an degree in mathematics. She is experienced in academic cryptography, but little else.

Here is Azal's estimate:

Learning Objective	Value	Learning	Learning Objective	Value	Learning
Network types and topologies	$50	0%	Symmetric Cryptography	$50	100%
VOIP	$75	20%	Asymmetric Cryptography	$50	100%
TCP, UDP, ICMP	$100	20%	Hashing	$75	100%
DNS	$100	50%	VPNs	$100	25%
TCPdump	$25	20%	PKI	$50	100%
Routing Protocols	$90	50%	Steganography	$50	100%
Physical Security	$20	75%	PGP	$50	100%
Technical Controls	$75	50%	Wireless	$75	25%
			Legal Issues	$25	50%

And here is Morgaine's:

Learning Objective	Value	Learning	Learning Objective	Value	Learning
Network types and topologies	$100	100%	Symmetric Cryptography	$100	10%
VOIP	$50	100%	Asymmetric Cryptography	$100	10%
TCP, UDP, ICMP	$70	100%	Hashing	$100	10%
DNS	$100	100%	VPNs	$75	80%
TCPdump	$50	100%	PKI	$100	20%
Routing Protocols	$50	100%	Steganography	$50	50%
Physical Security	$50	100%	PGP	$100	50%
Technical Controls	$50	100%	Wireless	$100	100%
			Legal Issues	$100	100%

Azal and Morgaine scored both value and learning differently. When estimating a market value for a skill, you're going to score what you already know higher simply because it's more familiar. If you're concerned with accuracy, have others review your estimates to bring the value more in line with reality. Your perceived percentage left to learn will also have a familiarity bias. Beginners often feel that the more they know, the less they have to learn. With experts, this tendency reverses itself as the more they know, the more they know there is to learn. Correcting for these biases is less critical if you are using the tool to choose which class to take or whether to learn on your own.

To calculate the value of a class, simply multiply each value by the amount left to learn and tally up the total score card. In this example, Azal estimates that days 1 and 4 would be worth $618.75 to him and Morgaine estimates them to be worth $905.00 to her. Suppose this class cost $750.00 (not true, but since we're just looking at two days from a six-day class, it's a reasonable simplification). Azal

would not get as much out of it as he'd be paying, so he should look at other classes or other modes of learning. Morgaine would clearly get the value of the class plus $155 more and should thus consider it.

For other classes, the approach is the same: just run through this exercise for every class that interests you and passes a basic quality check. The one that has the highest value to cost ratio is the one you should consider first. Although the value numbers you've assigned are arbitrary, if the same arbitrary values are used consistently, the relative values of the classes will still be correct.

SELF-STUDY

The other classic method to bootstrap learning is to go through self-study. While it is often the cheapest and fastest way to learn, self-study can be difficult if you have problems focusing, completing, and limiting interruptions. To be successful at self-study, you must be able to set a plan and stick to it. A common failure mode for this style of learning is to lose time doing things that do not advance your goal. When learning in a class environment, the instructor is responsible for the structure of setting your goals and helping you maintain your forward progress. Without an instructor, you are on your own.

Many books and recorded lectures are available for much of information security, from which you can get some structure. Most "learn on your own" books and recorded lectures function much like a class, where you are presumed to start with a certain level of knowledge and gain more understanding with each portion that you complete. At the end, it is expected, you have complete knowledge and are done. However, unlike traditional classes, this approach has at best only self-assessment tests and quizzes, so you don't know what you've actually learned. This mode of self-study requires more rigor and self-honesty to be successful.

Not all learning materials, books, and recorded lectures will be useful to you, or worthwhile in general. Read reviews, check sources, ask around, especially your mentor, and fact check the material before investing much time or money.

Information technology and information security computer aided training/instruction (CAT/CAI) is almost uniformly poor. If you have free access, give it a try, especially for introductory topics, but seriously reconsider paying any money for it. However, if you have skills in curriculum development, and authoring tools then good information security CAT/CAI would be a valuable contribution to the community. See the Chapter 4.1 "Boosting—Author" and Chapter 4.7 "Boosting—Community Support" chapters.

If you are not taking a class-based approach to your learning, it is still wise to lay out a rough plan. You may know what you want to learn. This should help you design a "final exam" so you can verify what you learned. The goal may be something technical, such as setting up an Apache web server to perform a specific goal. It may be nontechnical, such as writing a report or paper for someone else to read. It may even be externally measured, such as passing a certification exam. Only by having a goal in mind, and a means of testing, can you truly call what you're doing self-study.

To be highly successful (as defined by learning a large amount in a relatively short period of time), you also have to track your days and weeks of study. These metrics help keep you from following dead-ends and tangents into areas that matter little to your end goal. One way to do this is to break apart your learning process into smaller pieces. Each piece should be roughly the same size—about a day's or a week's worth of work. Each piece should also have a set test at the end, so you can verify that you learned it sufficiently well to move along to the next item. Once you have your agenda down, you can start the self-study process.

PROJECTS/EXPERIMENTATION

If self-study doesn't work for you, either because you have difficulty learning that way or because you face constant interruption, you may learn best by doing. Many people learn best by blending the Learn and Do phases. However, if you learn by doing, your first work will necessarily be incomplete and flawed. The old programmer's adage of "write one to throw away" really applies well here, so plan to do more Learning and Doing.

Fundamentally, the issue is that people often learn best through mistakes. We typically learn from the mistakes of others; we're hard-wired for story, and stories of wild successes and massive failures are what stick with us. However, if we're blazing new ground or have to learn on our own, we must make our own mistakes.

There's nothing wrong with this. Scientific studies have shown that there is a measurement, called Error Positivity (EP), that shows how well people learn from mistakes. Those that have higher EP tend to learn more and better than those that do not. Since high EP correlates to positive attitude and a willingness to work but does not correlate to intelligence, there is a theory that this is why, over time, competent hard workers outperform their more intelligent but less hard-working competition.

However, if your learning process was built out of a series of mistakes, it's likely to not be sufficiently stable to build upon. In technology in general, we tend to build something until it works and then move on. This makes sense because it's a lot easier to determine when a thing is functional than when it is secure. However, if you want to break into the information security field, you have to build things that work, but also break them until you understand how they fail. This is a very different process and, as you're learning, can be dangerous because by definition you don't know what you're doing.

People who learn best by doing often do it while working for someone else. This may be on-the-job learning or learning done on your own but involving work-related systems. This may be work done when volunteering for nonprofit groups or open-source communities. But learning while doing is a risk to your company, to your customers, to your friends and to yourself.

Stop and ask yourself, "What did I learn from this?" and "How can what I built harm others?" and "How do I prevent harm?" Apply the answers to these questions and fix the mistakes you made as you went. In some cases, this may require a complete rebuild; in others, it may involve leveraging other skills you have to harden the environment that people will be using.

However you decide to minimize the risk to you and others, remember that security is about protection. It's also about helping some people achieve their goals, while limiting attackers' options. However, at the end of the day, if what you did doesn't improve the protection of someone else, you can't claim that it was security-related work.

BREAKING DOWN TO BREAK IN

The rest of this book focuses on specific jobs, roles, and tasks that are common in the information security industry. Each section within the Learning tier details a specific entry-level job—what it's like and how to get into it. We will detail not only the duties of the job, but also how and why the job might be less than enjoyable. Entry-level jobs are by definition stepping stones to something greater, but where you go is often a reflection of where you have been, so jobs that you hate will taint every job after as you'll be following a path based on work you dislike. Some paths will not be worthy of your goals.

To get where you want to go, you need to know where you are and what additional skills you need to acquire to get there. There is always more than one way to do something, and that is true for career management as well. You'll find some jobs to be dead-ends, where it feels as though there is no way forward. Jobs based on dying technologies may result in your feeling as if the market has dried up.

Remember, however, that your actual job is not reviewing firewall logs or deploying patches. It's protecting people. So long as you keep that in mind, you will find other ways to protect people. Maybe it will be with a new technology. Maybe it will be replacing the legacy technology in which you are currently an expert with newer technology. Maybe it will be something nontechnical, and you'll move into management.

As long as people are cheap, lazy, or stupid, you'll have work. The trick—the only trick, really—is not to be cheap, lazy, or stupid yourself. Fortunately, the opposites: thrift, hard work, and intelligence also provide work, and will usually be more pleasant. Learn about new technologies. Invest in and develop new skills. Hold yourself to high standards. Constantly improve. And, finally, always test your understanding. Do all these things with each job and, as you move through your career, you'll find the job you want and how to get it.

CHAPTER

TIER 1—LOG REVIEWER 1.1

INTRODUCTION

> "Once is happenstance. Twice is coincidence. The third time it's enemy action."
>
> **— Auric Goldfinger in *Goldfinger* by Ian Fleming, 01959**

The log reviewer role involves a periodic review of files that store critical data about what happens within your environment. These logs typically originate from applications or from devices like firewalls or servers. In more mature organizations, the logs will likely be stored centrally in a Security Information and Event Management (SIEM) system. These logs typically contain an ongoing description of what the system is currently doing—indicating whether the system started up properly, which specific events occurred, what problems were encountered, what failed, and finally whether the system shut down properly.

In many organizations, this role is entry-level and may be filled by untrained individuals or those in a junior role in system or network administration. In larger organizations, this is a full-time job that may be one of the roles in a dedicated team. In very well-developed security organizations, this team may be dedicated to a centralized (politically, if not physically) Security Operations Center (SOC) continuous 24/7/366 real-time monitoring and response. A well-developed information security organization will have a full staffed SOC with enough log reviewers to allow all critical security logs to be reviewed for all operating hours. Although "SOC" is a common information security term, it is not universal, and even very large organizations may not have a SOC by that name, or any single unit that does SOC functions. A network operations center (NOC) may include some SOC functions. In these cases, the term "SOC" represents the job functions, if not the organizational unit itself. It also represents a possible opportunity for an ambitious and resourceful person who could create a SOC where none previously existed. Lacking a SOC is also an opportunity for attackers and represents a risk to the organization.

At a technical level, most people never interact with logs or even know they exist. Logging is often an afterthought, even with information security systems. However, the logs can be a valuable way to troubleshoot a system, both during an incident and after the fact.

There tend to be two types of logs: debugging/informational and security. Information logs are used to track application behavior and troubleshoot; most developers and system administrators are used to working with them. Security logs are different. These logs store critical information about system use, such as when people log in and out of systems, when people fail to log in successfully, and what data access attempts fail or succeed based on access controls. These logs, such as audit logs and security

event logs, can provide a great deal of information about security issues on a system, which is why attackers often alter or remove logs to hide and destroy evidence. Network security logs in particular can track which websites people visit and what sorts of network applications they run.

The job of a log reviewer is to ensure that any indications of attacks are detected and responded to within a reasonable amount of time where "reasonable" can include immediately. Logs need to be regularly reviewed, perhaps even in real-time. In addition, log reviewers may be responsible for managing and protecting logs so they are not not viewed by unauthorized people or modified in any way. The job can be boring, but it is also absolutely essential.

HOW TO BREAK IN

Log reviewing is often extremely boring, so it's often the first thing offloaded onto new or junior staff. Though logs are often dull, they can hide gems. It can be interesting to trace a 20-year-old bug as you analyze network traffic from one system to another to identify why a particular log entry keeps recurring.

Reviewing logs can be hazardous. You may discover information you shouldn't know, such as possible acquisition targets, just from seeing the browsing habits of senior managers. You may discover activities that shouldn't be occurring, such as employees selling illegal items or browsing to online gambling sites. Of course, you may also identify indications of compromise (IOC) in the attack's kill-chain that can help your organization get in front of an attack and determine how best to respond.

The role requires being trustworthy enough to not snoop where you shouldn't, but also intelligent enough to see and investigate things that look odd, or hinky. This is a great opportunity to learn. How you break in will depend on the sort of business you wish to target.

An otherwise fully staffed SOC may have positions available for those that start out in log reviewing. A SOC may require employees to have backgrounds in the systems they monitor, including non-security systems. These may include systems that support business operations, software development, system administration, accounting, customer support, and even marketing. Working in these areas can bring useful skills, knowledge, and business contacts to a SOC. Your experience in these non-security areas and understanding the technical underpinnings for how these systems function can help land your job in the SOC.

Smaller firms will not have a SOC, but log review is still required. This may be part-time and involve reviewing each system individually or looking at centralized SIEM log reports. If there isn't a centralized SIEM, your best way to break into the role is to spend some time implementing a central SIEM logging system and then tune it. If you are starting from scratch, look at the Security Onion, a free Linux-based system created and supported by Doug Burks. See the Appendix: Tools for details.

HAZARD WARNING

Policy Compliance

Activities such as setting up a centralized log service, any kind of monitoring, or any kind of server may have serious policy or political issues, and as mentioned above, moral issues. Make sure that such activities are within policy, within your job description, and known of and approved by your management in writing. Management approval needs to include conflict of interest resolution for monitoring of co-workers and direct management..

In a small organization a centralized SIEM system can be implemented on old hardware that may be lying around and, as you turn on the SIEM and tune it, you will find many internal activities that would be of interest to your boss. Tuning any alerts that occur may involve touching many different systems and implementing changes to things like web filtering and MS-Windows log settings, and removing unnecessary services on numerous servers.

If you wish to move to a new company in such a role, you will need experience. You can gain such experience from implementing a SIEM on your own, perhaps at a nonprofit or school near you, or your own home environment if you can't get approval elsewhere. This not only gives you the skill you need to get over the "need experience to get experience" hump, but also gives you an excellent story to tell in interviews.

HOW TO IMPROVE YOUR SKILLS

The contents of security logs can be highly counterintuitive. A sternly worded log warning message about dangerous activity could be an indication of compromise (IOC) or be a completely normal, but misleading, entry. Simply by looking at logs, real-time, or historical, a log reviewer will begin to understand what is common and what is not. Note that commonly occurring activities may actually be quite dangerous—just because it's frequent doesn't mean it should be occurring. Log reviewers learn the difference both for the general case, and for the specific environment they are reviewing. Context is critical.

By comparing log activity with known events, and by comparing logs of different systems, the log reviewer can start to build the skill of understanding the environment. A seasoned Log Reviewer will understand what a particular log entry really means, and if it's normal and innocuous, or subtle warning of serious security issues.

There are several areas in which you can grow your skills as a log reviewer.

The first, and most obvious, is tuning the logs within the logging system. Whether you're just using a standard syslog server or a much more complex enterprise log management system the system will need tuning. A tuned security logging system will log everything that is needed, log nothing that is not needed, and alarm, alert, and/or merely warn where needed. Every time a new server comes online or a major upgrade is installed, the logging system will detune and the logs will become "chatty." The trick behind tuning is to make a choice as to whether to adjust the source of the logs or the target.

For example, the log source may be a server that is running in debug mode because a developer needed some information, perhaps a lot more information. This mode is seldom turned off, but just resetting the log level to "informational" may reduce the logging to a more manageable level.

At the log target side, you may find that every Monday morning a legacy system sends ten thousand otherwise innocuous packets that trigger an alert. Legacy systems are notoriously difficult to change, but once you know why something is happening, if you can't stop it from happening, you can instead adjust the alerting system so that it knows that the problem isn't serious and shouldn't alert.

Another way to grow your skills as a log reviewer is to look at any custom logs that a developer may have implemented. These logs may contain extremely sensitive data, such as passwords or Social Security numbers, so you could get involved with developing a tokenization or masking system so the sensitive data is not compromised, but the logs remain useful. This is also an opportunity to learn software development and further expand business contacts.

Yet another way to grow your skills is to expand your logging capabilities. Many organizations are either running a central log collector or a network-based monitor. Usually only larger and more mature organizations do both. Expanding from network utilization and service monitoring to alerting on security events is often a very easy step to take. Implementing a log parsing and alerting system is also often something you can do to improve your position.

Finally, integrating trusted alerts into your ticketing system can improve efficiency and demonstrates a solid understanding of your environment.

RECOGNIZING WHEN YOU'RE STUCK

Security logs usually have recurring patterns, both short- and long-term, ranging from seconds to months. Some environments can take years to really understand, due to seasonal and yearly fluctuations. Although computers don't have seasons, the systems they are a part of, such as retail businesses, schools, and sports, can be directly affected by seasons.

Organizations often consider log review to be a low priority task not requiring a dedicated person or team. In these cases, log review might be even ignored and replaced by other tasks considered more urgent. This is an opportunity to move over to these other areas full-time. An organization that doesn't consider security log review and monitoring important is quite possibly an organization to learn from and then leave behind. Some organizations do consider security log review important enough to do consistently, but the organization may still not provide an advancement path.

After you've been through a few seasons of logs review you should have been able to grow your skills as mentioned above, but if not, then it may be time to move on.

HOW TO GET OUT

Log Reviewer may bring skills and knowledge into new areas. For example, by focusing on particular issues, a Log Reviewer may have valuable insight into Coding/Development, Patch Management, System Administration, and Network Administration. In some organizations Log Review may be so important that it requires a Subject Matter Expert.

This deep knowledge can be used to more quickly evaluate events and issues, and to start automating log review tasks.

1. Do: In a quiet, low-threat environment, a log reviewer may be able to quickly determine that no issues exist for the moment, so they can start growing skills in related areas.
2. Do: Partially automate the log review task with simple filters (Chapter 1.4 Coder/Developer).
3. Do: Fully automate with more complicated filters and custom code and reports (Chapter 1.4 Coder/Developer).
4. Learn: Dig deeper and learn more about the logs; notify what systems produce those logs (Chapter 1.5 System Administrator).
5. Teach: Find non-security issues in the logs and notify and help others in the organization (Chapter 1.9 Quality Tester).
6. Teach others how to be a log reviewer, as someone else will have to do it when you leave (Chapter 3.0 Teach, Mentoring section).

7. Volunteer at work or do personal activity for specific tasks outside of pure log review (Chapter 4.0 Boosting).
8. Write an article, or do a presentation on what you've done, learned, and taught.
9. Advance: Automate log review to the point you are no longer needed. Your new job, perhaps at a new company, won't be log reviewer, but instead will be a multi-classed log automation specialist, which combines log review and coder/developer and quite possibly adds system administrator and network administrator.

CRITICAL WARNINGS

Unlike many information security roles, the stress level of a Log Reviewer is simultaneously and paradoxically both high and low. Commonly described as "long periods of boredom punctuated by moments of sheer terror," the life of a Log Reviewer will have its up and downs. Most environments are relatively low in terms of threats, and most threats are not subtle, so the stress is usually low. However, one anomalous event may occur that will spur further investigation, causing stress levels to spike to extremely high levels. That the high stress can occur at any time without warning is itself stressful.

Log Reviewer can be a dead-end job and should be avoided by those who don't have the circumstances, patience, or discipline to benefit from its strong learning opportunities and potentially long wait for advancement.

Table 1.1 Role at a Glance—Log Reviewer

Hours	Travel	Stress Level	Creativity	Flexibility	Stability
0.5–8 hours/day	Usually none	Generally low	Low	High	High
General job duties	Reviewing logs. Writing log filter rules. Fulfilling log research requests.				
Learning	High – Reviewing the logs for an entire organization covers all aspects of their operations. You can learn about how firewalls work from the firewall logs and these logs potentially contain the details of all Internet access by everyone in the organization and details on everything on the Internet that connects to that organization. Anti-virus logs contain details on what malware the organization encounters and how users react to it. As you troubleshoot issues, you may find yourself learning about servers, networking systems and vulnerability management. The learning opportunity in this role is very high, but you will likely have to push your way into the learning as the role is often also culturally isolated and time limited.				
Advancement	Typically, people advance into the role as an entry level position or laterally from junior System Administrator or Network Administrator. Expect to advance to Security Assessment, Risk Assessment, and especially Incident Responder. Log Review can also serve as critical experience for Incident Responder, Security Architect, Security Consultant, or Security Management. Culturally, most organizations will provide little in the way of formal advancement paths for Log Reviewers. If you serve within a centralized Security Operations Center (SOC), you may be able to become a team manager, but generally speaking, you will have to either change employers or petition a superior to move out of the role.				

CHAPTER

1.2 TIER 1—PATCH MANAGEMENT

INTRODUCTION

> "All programmers are optimists."
>
> **— Fredrick P. Brooks, Jr.**

Patch management is a component of configuration management where you are responsible for ensuring that patches are applied to computers as directed by a company's policies and procedures. Generally, when a patch is released, individuals in Vulnerability Management roles will identify the urgency of releasing the patch based on the vulnerabilities it addresses and the systems to which it applies. Once the patches that need to be released are identified, those in Patch Management are responsible for ensuring that the patches are tested before release, if possible, and then applying the patches to the appropriate systems.

Depending on the devices being patched, various tools may be used to manage the process, such as Microsoft's System Center Configuration Manager (SCCM) and SolarWinds Patch Manager. You may be responsible for a single platform (such as those workstations running Microsoft Windows) or multiple platforms.

This is the type of job that can be very monotonous in its duties but, if you take some initiative, it can provide you with many opportunities for learning.

HOW TO BREAK IN

Though technical skills are important, being well-organized and comfortable following set procedures is at least as important for this role. If there are no available mentors you will need a clear understanding of the organization's platform and environment. Other organizations may have mentors who will train you into the role if you are able to demonstrate solid general technical skills.

In some organizations, this role can be used as an entry point into the IT field. There are many standardized tasks, and the primary job responsibilities are to monitor, document, and report, so interns or other entry-level people may be assigned this work under close supervision. People who are already enrolled in or have graduated from a computer-related degree program are often given preference, since it demonstrates an existing level of general technical skills.

For non-intern positions, showing how you provided support as a volunteer, or helped patch computers at a local nonprofit, are ways you can demonstrate experience in this area.

HOW TO IMPROVE YOUR SKILLS

Some people view patch management as a dry and boring field, but if you are willing to put in some extra effort it can be hugely beneficial for learning about the information security field. Nearly every patch addresses some form of security vulnerability—something that either has been, or could be, used to infiltrate systems.

Take the time, either at work or on your own, to deeply examine the patch and the vulnerabilities they address; they are a great repository of security knowledge. Understand not only the specific vulnerability that is being addressed, but the class of vulnerabilities it falls under, what causes the vulnerabilities, and how to prevent them. Learning the difference between a buffer overflow and SQL injection is important for the security professional. Early on, it is unlikely that you will be able to do so for every patch, but as you go on and develop your knowledge, it will become easier and easier. This same technique is used by penetration testers and security researchers to develop attacks against un-patched systems and to develop brand new attacks. Patches often only fix a specific vulnerability, not the class of vulnerabilities.

As you get better at understanding what sorts of attacks a patch prevents, you will also become better at implementing such attacks yourself, should you wish to eventually move into active auditing and penetration testing of systems.

Similarly, one of the key concepts of security is understanding that security is not about technical solutions addressing technical problems; security is about reducing risk to a level with which the business is comfortable. Take the time to learn why some patches are critical to your organization and others are not and you will learn the business drivers for security. Learn the different reasons that may make one buffer overflow patch critical for immediate installation, outside of the normal patch schedule, while another patch can easily be installed in the regular patch schedule, or not at all.

Well-run organizations will have processes for testing patches before their rollout. Take advantage of the fact that these processes have already been set up to learn about them. Identify the key characteristics of your organization's processes, and work to learn enough about them that you could implement your own elsewhere, should it become necessary.

Finally, take the time to learn about your organization's platform and environment.

RECOGNIZING WHEN YOU'RE STUCK

- You dread Patch Tuesday.
- You fear Exploit Wednesday even more.
- You don't care if that patch really did get applied.
- You don't get a sense of job satisfaction from having the best-patched environment you can manage.

HOW TO GET OUT

These roles are common at large organizations, so if you are looking to stay in the same role, you will usually be able to find a similar role elsewhere. Server Administration roles could be an advancement opportunity, as could a Security Coordinator role. If your organization splits patch management by platform, this could be an opportunity to switch platforms and retain your patch management

experience. Alternatively, if your organization splits patches by internal organization structure this can be an opportunity to change job role and retain your platform experience. Other organizations split patches between servers and clients, or operating system and applications. There is no one best way, only what works for that organization.

If you have taken the time to learn deeply about security throughout your time in Patch Management, then advancement to Vulnerability Management, Auditor, or Security Assessment, depending on the skills that you have developed along the way, may be possible.

CRITICAL WARNINGS

If you are in this role too long, you will have a hard time leaving it. Patch management is often underfunded. Ideally, this role would involve you building your skills and automating patch processes within a test or lab environment. Then, when you have enough skills to be useful elsewhere, you get promoted and another person would come in and take over your role.

Sadly, this is often not the case. Many organizations don't have test environments. All production environments should have dedicated and safe test and experimentation environments. Patches can break a system, make it unstable, or take away key functionality. Sometimes key functionality is a security vulnerability and therefore a required feature! If an organization is not willing to spend the time and money to address these risks, you should be wary of staying there too long, as they are not supportive of helping your efforts to improve things.

Also, since it is an entry-level role, you should be working to get out of this role in one to three years. The organization and team you end up in will have a significant impact on your ability to grow. If you are left in your cube, only asked to perform your tasks, and never provided opportunities for cross-training and growth, that is another sign that the company is not serious and you probably need to plan to build skills on your own so you can move on to a better job sooner rather than later.

Table 1.2 Role at a Glance—Patch Management

Hours	Travel	Stress Level	Creativity	Flexibility	Stability
8 hours/day Overtime	Low to None	Low	Low	Low	High
General job duties	Testing patches on non-production systems. Setting up systems for automatic distribution of patches. Documenting the application of patches.				
Learning	Medium High. Leverage platform skill to security management skills. Leverage security skill to platform skills. Get exposed to vulnerability management and incident response. Many opportunities for the driven individual. Can be effectively avoided if you so desire, but don't do that.				
Advancement	Vulnerability Management, Auditor, Security Assessment or Incident Responder, depending on the skills that you have developed along the way.				

CHAPTER

TIER 1—HELP DESK 1.3

INTRODUCTION

> "Don't Panic."
>
> **— Douglas Adams**

The Help Desk is a common part of larger information technology environments. It provides a single physical place, phone number, email address, or other communications channel for what are typically the most basic and least experienced information technology people in the company. Consequently, it's one of the most likely places an inexperienced person will start.

The help desk is where a "people person" is appreciated, but not always valued.

Some organizations separate the information security help desks from the information technology help desks, while others combine the functions. Either approach affords a good way to break into information security for someone with limited skills or experience. If you have a choice, the information security help desk might be a better place to start. But since information security is as much about the technology as about the security, an information technology help desk could also be a good start.

Consider these differences when choosing between a job at an information security help desk and an information technology help desk: Talk to people who work specifically in those areas in those organizations, talk to both your potential co-workers and your potential management. Find out how technical each is, how much flexibility in the job exists, and what you can learn, and where you can advance.

The stress level associated with this job is highly dependent upon the stress level of the incoming requests. For example, a help desk for a data recovery company may have highly stressed customers, so much that some data recovery companies provide specialized data loss bereavement counselors. Additional stress comes from help desk operators having limited to no ability to fix systemic problems, which will be highly stressful when there are many systemic problems.

Pay is usually hourly; work hours can be highly variable, and may include working nights, weekends, and holidays, yet still be part time and seasonal. Overtime is rare.

Help Desk employees usually operate in fixed shifts, but not necessarily as part of regular 8-hour days. Hourly employees are often limited to 40 hours a week or less, while seasonal variations, and extended operating hours may result in irregular hours, but which may provide high work hour flexibility to help desk operators.

Work flexibility is usually stable, but some organizations have seasonal workforce fluctuations. The Help Desk tends to be the largest group with the lowest seniority and least experienced staff, which may result in less stability.

Help Desk duties tend to be boring, with the same types of recurring customer problems—sometimes exactly the same problems, which becomes tedious, and possibly also stressful. The most interesting problems can't be solved at the lowest help desk tier and are moved up the chain to more experienced staff.

HOW TO BREAK IN

Help Desk is an entry-level position, and so is often a new hire position, perhaps an internship. However, in some cases it may be a horizontal movement from a non-technical or non-IT position. Non-technology companies may provide this horizontal movement for those who are interested in help desk jobs, and may also provide training. Someone with some basic technical skills but not enough to get hired directly may gain both experience and business contacts by becoming the local "computer person" that people go to before they call the official Help Desk. In some environments they can immediately provide better support because Corporate Help Desk employees are unfortunately notorious for providing poor service, with long waits, poor people skills, or irrelevant advice. Becoming the local technical "go to" person can develop from any non-technical position; such a person has the advantage of already knowing the organization.

HOW TO IMPROVE YOUR SKILLS

Most help desks have frequently asked questions and problems. Not all help desks have detailed and up to date documentation for those questions and problems. Some have no documentation, or even training. Either way, it's an opportunity to Learn/Do/Teach. Read the documentation, if any. Ask questions. Research and get the answers to your own questions and those asked of you. Document and solve problems. Update and write new documentation using what you've learned. Teach what you know to your co-workers and your customers. If you're starting from nothing then as you take calls, do metrics, take note of how long calls last, and what the most frequent problems and questions are. Set up a documentation environment; share with your co-workers.

In some cases a technology company may put senior people on help desk duty to supplement staff. These are opportunities for you to learn from experienced people and develop professional relationships. The help desk environment can be a highly structured and stressful one, with limited time to do anything but take calls and resolve trouble tickets. If this is the case, use this as an opportunity to ask others, senior or not, how they learned to cope with the environment. If it is not, you should cultivate relationships outside of the organization and build a team of people of which you can ask general questions. You will have to be careful not to let slip sensitive information, but a well selected team of colleagues can be very helpful.

RECOGNIZING WHEN YOU'RE STUCK

An organization is stuck when the same problems keep recurring which could be fixed with a better product or at least better documentation. A good Help Desk environment treats help desk tickets, resolved or not, as information for making better products and services. If the organization isn't learning from this, and you as a help desk operator aren't either, it's time to move on.

If you're learning and becoming a senior person, but you haven't had at least a minor promotion or had your hard work acknowledged in six months to a year of starting, it might be time to move on. Either the organization is using you for cheap labor, or you aren't working out in that environment. This doesn't mean you aren't suited for help desk, rather that that particular work situation may not be good for anyone, or only for certain people. Many people choose to move out of the help desk environment when the opportunity arises.

If you find yourself well suited to the role, you may find yourself promoted to Help Desk Lead and, eventually, into management. This is not a way to break in to information security, but it can be a very satisfying career nonetheless.

HOW TO GET OUT

Entry level Help Desk is close to the bottommost tier in the support structure, just above the end user. They are the ones who directly handle calls and requests, while more skilled or experienced help desk people will often be moved up to a higher tier. Each tier hands off work they cannot complete to the next higher tier. To move up a tier doesn't mean being able to handle all incoming requests without having to pass on to the next tier; it means being referred by others who are at a lower tier.

Table 1.3 Role at a Glance—Help Desk

Hours	Travel	Stress Level	Creativity	Flexibility	Stability
20–40 hours/day, variable	Usually none, although may have different job sites in the same area.	Variable	Low	High	High
General job duties	Directly answering the phone. Responding to email. Working trouble tickets.				
Learning	Variable. Help Desk operators may have a formal training program, either on the job, or a classroom type environment. Supervisors are often also trainers, and sometimes even mentor-like. Every incoming request could be a learning opportunity and will be for the new help desk operator. However, over time, perhaps quickly, the learning opportunities tend to drop and the same nearly identical requests keep arriving.				
Advancement	Help Desks can vary significantly. In larger environments the Help Desk supervisor positions may be available as a growth position. The Help Desk may have significant internal and external turn-over, which can be an opportunity for the new help desk operator to stay and retain. In some organizations they are considered the menial labor of the information technology organization. However the Help Desk can be an entry level position for an information technology career. Some organizations integrate the Help Desk and use it to find and feed talent into the entire support organization. A good organization will see talented and hardworking operators and want to retain them. In not so good organizations a help desk operator will have to figure this out on their own.				

CHAPTER

1.3.1 TIER 1—HELP DESK—STORY

JIM CHAN

I started my IT career in 01999 by attending Brown Institute in Mendota Heights. I finished their program with an A+ certification and landed a desktop support job at Target Corporation. During a hardware refresh project for the Finance executives, my boss asked me to work with the Information Security team to ensure the drives were erased properly. Target Corporation is huge on mentorship programs and development plans. I checked with my boss to see if I could schedule an informational interview with someone on the InfoSec team. My boss gave me the name of the Compliance and Monitoring manager and I scheduled an informational interview. During this informational interview, the Compliance and Monitoring manager said that they have an internship program. My current boss approved, and the internship program was a one-day-a-week shadow and Q&A session with various members of the Information Security team. This is where I learned an overview of IDS, Forensics, Architecture, Identity Management, etc. At the same time, I started attending night school to finish up my bachelor's degree. After finishing the internship program, there were no openings in the Information Security team. I continued the mentorship meetings and started attending open weekly "cryptol-unch" lunch meetings with the Information Security team. After a few months, my mentor informed me of an entry-level user access management position on the Information Security team. I had finished my bachelor's degree before the opening came about, applied, interviewed, and got hired on.

The interpersonal relationships I developed through the internship program, having a bachelor's degree completed, and proactively getting involved with Information Security tasks and projects were the biggest contributors to breaking into the field.

CHAPTER

TIER 1—CODER/DEVELOPER 1.4

INTRODUCTION

> "What I really need is a droid who understands the binary language of moisture vaporators."
>
> **— Uncle Owen**

Everyone who worked with computers was once expected to program them. Now it is common for people to work in information technology without learning any programming languages at all. In this section we will discuss those few remaining people who start out in the development world and then wish to break into information security.

If you are approaching information security from this perspective, you should have some familiarity with programming languages. Once you get in, you may find yourself reviewing existing code for security concerns. You may be creating brand-new programs or extending old ones. You may have a bridge role between non-security developers and the information security teams. Most information security roles involve some sort of bridging, but few roles bridge more different worlds than between non-security developers and information security.

This bridge role role will probably be at most 75% development and the rest discussions, explanations, and meetings. You should expect to gain deep skills in languages and to get very good at understanding how other people communicate, identifying application and business model issues, and prioritizing tasks.

HOW TO BREAK IN—PRELIMINARIES

Typically this role involves first knowing one language very well. A common but bad practice is that good security coder/developers must know a little bit about many different languages. Commonly chosen languages include: C, C ++, Java, Javascript, Perl, Python, and Ruby. Knowing a little about each means your work will be poor in all of them.

There are no shortcuts to learning, and this applies to learning programming languages just as much as other subjects. Perhaps you spend one month learning Perl. Then, when you hit limitations to the language, you spend a month learning Python. As you dig into security scripts, you learn some Ruby. Later, you may create modules to get your code to run efficiently, which means two to three months learning C and another two months in learning C ++. In a year of learning, you've learned only part of six to eight languages but none of them very well, and perhaps not well at all. In a worst case, you may

only be able to write bad but otherwise useful code only by using all of the languages you know, instead of using the one language that would be best suited for the task. This is worse than the feared "spaghetti code," it's random layered pasta code.

Contrast this approach with the "learn one language very well, then move on" approach. The better you know your first language, the easier it is to pick up similar ones. Consider how much easier it is for English speakers to learn French and Italian after having thoroughly learned Spanish. The grammar, vocabulary, and logical structures of the latter languages are all similar, so picking up the differences is easier than learning any one of them from scratch. The same applies to programming. Once you learn one programming language thoroughly, you can use that knowledge to understand other programming languages. Documentation is easier to understand, because you've picked up the vocabulary around the language and can find what you need much more quickly. Troubleshooting code also becomes easier because you can quickly create proof-of-concept functions in first language to identify whether the issue is related to a bug or fundamental misunderstanding.

In general, you should start with the language you know best and spend upwards of six to eight months learning it as well as you possibly can. If you learn best from books (likely, as you're reading this one), read one or two beginner books, three to four intermediate books, and at least one advanced book. Select books with exercises to do, and never skip the exercises. The goal isn't to just read the book, the goal is to learn, and completing the exercises is your proof to yourself that you learned. The completed exercises are also a valuable reference for yourself and can be part of the portfolio you can present to prospective employers. A good example of book learning is the approach of Perl book publisher O'Reilly and Associates. If you start with *Learning Perl*, then move on to *Programming Perl*, *Mastering Regular Expressions*, and *Perl Best Practices*, then wrap up with *Advanced Perl*, you can jump from beginner to master in very short order. See the Appendix on Perl for details. Once you've mastered one language, you can pick up any intermediate book on C, Python, or Ruby and bootstrap yourself to an experienced level very quickly.

The same approach applies if you're learning from online materials, videos, or a classroom. Make sure that, before every new lesson, you understand the previous lesson as much as you can to minimize time lost to misunderstanding.

HOW TO BREAK IN—BEYOND THE BASICS

Once you have your basic skills down, it's time to leverage them and move in an information security direction. One common method is to join the team as a code assessor. These jobs are frequently advertised; in large companies they should be easy to find. In smaller organizations and some development teams, you may find a hybrid developer/security team lead role. For both of these, the basic approach is to apply for the position and then do well in the interview. This is, of course, easier said than done.

If you want to help a company start a development security practice, you have to come across as an expert. For this, it may help to develop some additional skill via one of the "boosting" paths near the end of the book. In general, working on a code similar to your target environment will do more for you than anything else. If you're targeting a company with legacy Microsoft ASP.NET code that it's converting into Java, you can quickly develop some experience. Find an older orphaned project written in ASP.NET that still has some legitimate value in terms of business logic on Microsoft's CodePlex.com, SourceForge.net, or similar open source hosting site. Then port it over to Java, documenting what

you did and why you did it. This builds your portfolio and will give you the type of story you'll want to talk about in the interview, greatly increasing your chances of landing the job. Increasingly developer resumes are their open source code repositories.

Unfortunately, most hiring firms are obsessed with specific languages, and if you don't have what they want, you'll be rejected. If you follow the above advice and know one language very well, you should be able to pick up your new target language very quickly. This can help you turn an industrial disadvantage into an advantage, since if they're discriminating against you for not having the target skills, they'll be doing the same to everyone else. If you can develop the target skills quickly, you can move into a much smaller pool of applicants.

There are a lot of programmers out there. The number of those programmers that know information security is considerably smaller. If you can target a specific language at an organization where you want to work, you've cut the smaller pool into a tiny fraction, and increased your chances. For specific uncommon languages like Haskell and Smalltalk, you may be down to a pool of one—which means you get to name your price for the job.

HOW TO IMPROVE YOUR SKILLS

Once you have the job, you'll have to stay on top of things. Unlike other information security jobs, you can't just stay on top of security events and new tools. You also have to watch for code libraries. A very common failing for developers is to use an older library and also fail to keep it up-to-date. This allows security vulnerabilities and incompatibility to creep into a code base that is completely isolated from any actively developed code. Thus, there are several vectors along which you will need to grow:

1. New languages—There is a tendency in development teams to use the newest or sexiest languages. This is part optimization, part planned obsolescence, but mostly fad and fashion. The optimization is related to the tendency, when learning a new language, to know the flaws of what you've been using but not yet encounter the flaws of the new language. Many shifts between Java and Microsoft.NET, MS-Windows and Linux, and Perl, Python, Ruby and Microsoft PowerShell are due to this "grass is greener" tendency. Your job, as a security expert, is to remain familiar enough with operating systems and languages to advise on the hidden information security costs of changing.
2. Defending against laziness—There is a saying that all programmers are lazy. The truth is that programmers want to program; they get into the field because they enjoy creating, but hate repetition. Programmers love automating, but hate what can't be automated. So programmers tend toward more enjoyable tasks like writing new code, learning new languages, but ignore other critical tasks like updating libraries and refactoring code to address noncritical flaws. They especially avoid documentation and development work needed to support fundamental infrastructure changes. Your job will be to advocate important security changes without alienating the team or being an irritant.
3. Ongoing analysis—Just as developers tend towards laziness, you must combat that tendency within yourself. You will need to create a schedule and stick to it, so activities such as pre-release code scans, team code reviews, and library assessment continue to be done properly. You also need to find and use as appropriate new security tools. For example, when input fuzzing was new a different fuzzer was released about once a month. Each tested for different issues, so each fuzzer

could have been run against each application, with each run resulting in more known security issues that would need to be prioritized for the future. The same idea applies to automated code scanners, vulnerability testers, and web application firewalls.

4. Programming—You must continue to improve your development skills. This can mean attending user group meetings, hackathons, and "code camps" within the community, going to official training, conducting your own training, or simply exploring a new area of the language. Many developers have achieved truly excellent levels by simply reviewing a different operator or function every single day, so that eventually they become world-class experts in the language.

RECOGNIZING WHEN YOU'RE STUCK

The most common place to get stuck in a coder/developer job is when the business stops valuing your contributions. In economic recessions, security is often the first to go as companies refocus their development efforts on adding features to attract new customers.

Though it sounds like a long time, you may find it takes a year or two to address the easiest findings. Some people make their entire careers moving from company to company, just working on this low hanging fruit. Others stay after the first set, and focus on next issues. Then there are the architects that typically come through last and address high-strategy issues such as framework and infrastructure changes, after which the first set of people is needed again. That the architects are involved last is one of the many ongoing issues in information security. Learn which phase you prefer, so you can identify when these corporate shifts happen and move to the next company.

You can also identify when you're stuck if you spend several months just addressing one class of issue in the code base. While there are some systems that need a year's worth of work to mitigate SQL injection issues, most developers choose to implement a global input validator and make sure that all classes of input types are handled there. The same approach can apply to other common validation issues, such as cross site scripting, request forgery, and the less common injection attack types like LDAP. If you spend too much time addressing a single class of flaw, it often indicates that your organization is operating in a break/fix mode and not making strategic progress. If you can't remove issues in the code faster than the other developers are putting them in, it's probably time to leave.

HOW TO GET OUT

There are many levels of work for a coder/developer. If you stay in this area, you may find a career's worth of work. However, if you want to shift focus, you can find an easy jump to system and network administration, as those roles are often made a lot easier with some scripting. At a more advanced level, programming skills will be useful in penetration testing and quality assurance. If you focus on code assessments, you can find a place on an assessment team focusing on internal code at various companies or departments.

Just as you improve your skills before you get your specific coder/developer job, you can do the same thing when jumping into another role. Find out what languages are likely to be needed in your future job and spend some time at your current job learning them. Again, once you have the basics down, you can pick up the new languages or frameworks you're going to need surprisingly quickly.