Making Everything Easier!™

Cobalt Special Edition

Crowdsourced Pen Testing

FOR DUMMIES®

A Wiley Brand

Learn to:

- Apply the key elements of a crowdsourced pen test
- Leverage the creative power of the elite crowd
- Accelerate remediation processes for pen test findings

Brought to you by

Cobalt

Caroline Wong
Mike Shema
Timothy L. Warner

About Cobalt

Cobalt's hacker-powered application security solution transforms yesterday's broken pen test model into a data-driven vulnerability management engine. Fueled by our global talent pool of trusted ethical hackers, Cobalt's SaaS crowdsourced pen test platform delivers actionable results that empower agile teams to pinpoint, track, and remediate software vulnerabilities. Hundreds of organizations now benefit from high-quality pen test findings, faster remediation times, and higher ROI for their pen test budget. Visit **www.cobalt.io** to learn how Cobalt is securing apps at the speed of business.

Crowdsourced Pen Testing

FOR DUMMIES®
A Wiley Brand

Cobalt Special Edition

by Caroline Wong, Mike Shema, and Timothy L. Warner

FOR DUMMIES®
A Wiley Brand

Crowdsourced Pen Testing For Dummies®, Cobalt Special Edition
Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-41607-4 (pbk); ISBN 978-1-119-41606-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact Branded Rights&Licenses@Wiley.com.

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Development Editor: Elizabeth Kuball
Copy Editor: Elizabeth Kuball
Acquisitions Editor: Amy Fandrei
Editorial Manager: Rev Mengle
Business Development Representative: Karen Hattan
Production Editor: Magesh Elangovan
Special Help: Jacob Hansen, Ante Gulam, Chris Tilton, Julie Kuhrt

Table of Contents

Introduction

Penetration tests and vulnerability assessments help illuminate, quantify, and qualify the bugs and flaws in a web application, mobile application, or application programming interface (API). Even though pen tests are typically facilitated by a security team, they should be treated as an essential, integrated part of an application's development life cycle. Not every application is compromised by techniques used in pen tests, but the security of every application can benefit from them.

Whether aligning pen tests with major feature releases or using them as periodic checkups, you can discover what kinds of vulnerabilities have slipped through your development process. Use a pen test to find vulnerabilities, reduce risk, and provide feedback for developers.

This book is about crowdsourced pen testing, a new approach to application security.

About This Book

Crowdsourced Pen Testing for Dummies consists of six short chapters that explore how crowdsourced penetration testing works and its defining characteristics. I begin by explaining the evolution of application security (Chapter 1), describe how a crowdsourced pen test works (Chapter 2), describe the benefits of a crowdsourced pen test platform for agile development and DevOps teams (Chapter 3), show you how to evaluate talent (Chapter 4), and explore some other options (Chapter 5). You walk away with valuable references for further study (Chapter 6).

Foolish Assumptions

I assume that you know a few things about application security. Perhaps you're a technical executive (CTO, CISO, VP of AppSec, VP of Development or DevOps), security team member (application security engineer or someone in an audit/compliance role), or an engineer involved in web, mobile, or API application development. As such, this book is written primarily for technical readers who know a little something about SQL injection and XSS and have heard of the OWASP project.

Icons Used in This Book

Throughout this book, I use special icons to call attention to important information. Here's what to expect:

This icon points out information that you should commit to memory, or at least have stored in your ready reference system.

This icon explains technical jargon in easy-to-understand terms.

This icon points out definitions, helpful suggestions, and other useful information nuggets.

This icon draws your attention to choices that could lead to bad results for you or your business if you're not careful.

Beyond the Book

If you find yourself at the end of this book, thinking, "Gosh, this was an amazing book. Where can I learn more?," just point your web browser to `https://blog.cobalt.io`.

Chapter 1

The Evolution of Application Security

In This Chapter

- Reviewing the three major developments in application security
- Understanding the recent industry trends that led to crowdsourced security
- Introducing the concept of a crowdsourced pen test

Application security is now one of the top spending areas for chief information security officers (CISOs), yet the same types of issues exist today as they did 10 or even 20 years ago. Why are the most elementary security vulnerabilities still showing up? The time has come to do something different.

Specifically, how can CISOs change the way that software applications are secured? How can they examine their habitual problem-solving approaches, learn from their mistakes, and move forward with a new and innovative approach?

In this chapter, I take a look at how application security has been applied in the past and discuss how you can do it better in the future. It's time for a radical change.

DevOps practitioners are not hindered by the past and have zero baggage when it comes to trying a new path. To quote the Director of DevOps from the leading online marketplace for real estate investing, "We use what works best and can match our agile processes, period."

Past, Present, and Future

The emergence of crowdsourced platforms will fundamentally alter the face of application security. It will impact the security, development, and operations professionals and the security researchers and hackers in the industry. Therefore, understanding the underlying trends is paramount.

In this section, I start by taking a step back and reviewing how application security has evolved. The industry has gone through three major developments, or waves (see Figure 1-1).

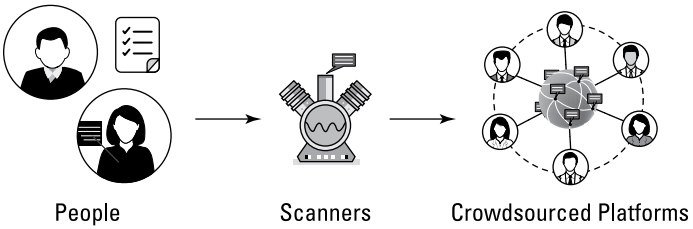


Figure 1-1: The three waves of application security are people, scanners, and crowdsourced platforms.

The first wave: People

Since the early days of application development, the need for security testing has been driven by the sharing of IT resources. Whenever multiple users are using a system, there is the risk of someone maliciously attacking other users of the system. Therefore, as time-sharing systems became a common computing paradigm in the 1970s, the first penetration tests emerged.

A *penetration test,* also called a *pen test,* is a formalized practice that evaluates systems, services, and applications with the goal of discovering security vulnerabilities. In security nomenclature, a *vulnerability* represents a weakness that can be exploited by a malicious individual, group, or computer process.

The first wave of the application security industry occurred in the 1990s as the foundation of the application security industry was established. One major driver for this wave was

the launch of the Mosaic browser in 1993, which resulted in an increase of personal computing and electronic-commerce. During this period, the first application security consulting firms, certifications, and conferences emerged.

The first wave centered on people specifically. Early hackers organized and launched nonprofits, companies, and conferences. With this came some of the first security companies such as IBM X-Force, FishNet, @Stake, and FoundStone.

The second wave: Vulnerability scanners

At the turn of the century, things started to change. Application security experts built commercial scanners to automatically discover vulnerabilities in web apps, scaling some parts of the manual effort that had been done in the first wave by humans.

A *vulnerability scanner* is a computer program that is designed to test and report on system security. The "system" tested by the vulnerability can be a single computer, multiple computers, network connectivity devices, applications, or services.

Over the past decade, application security vulnerability scanners have been organized into two categories:

- **Dynamic Application Security Testing (DAST) scanners:** These scanners perform automated tests of live, production-class web applications.
- **Static Application Security Testing (SAST) scanners:** These scanners focus specifically on an application's underlying source code.

Either scanning approach faces particular challenges that I discuss in Chapter 5.

The third wave: Crowdsourced platforms

The third wave of application security began with public bug bounty programs that introduced a way to connect organizations with freelance security researchers through a platform.

A *bug bounty* is a program instituted by a software developer or business that offers individuals benefits (perhaps recognition and compensation) for being the first to report valid security vulnerabilities and programming bugs in their software.

The first bug bounty program dates back to 1995, when Netscape offered cash for vulnerability reports against its web browser. However, it wasn't until about 15 years later that many more organizations started bug bounty programs. In 2010, Google expanded its bug bounty program to include its web properties. Shortly thereafter, Facebook and PayPal followed suit.

These bounty programs popularized crowdsourced application security. Between 2012 and 2013, the first bug bounty–focused startups — Bugcrowd, Zerocopter, HackerOne, Synack, and Cobalt — leveraged the crowdsourced model to great benefit for all involved.

In the third wave, businesses can direct and manage the creative thinking of globally sourced technical talent; that's the primary goal of a crowdsourced security platform.

Here are the major advantages of the crowdsourced model:

- Access to a global talent pool
- Managed findings (vulnerability reports) as a service
- On-demand scheduling

The third wave has fundamentally altered the face of application security. This transformation is not specific to bug bounty but will impact the application security industry as a whole.

Recent Industry Trends

A few major industry trends have come together at the same time, making the technology landscape ripe for crowdsourced application security:

- **Automation:** Today, many businesses design, deploy, and maintain their web applications and APIs by using a public cloud provider like Microsoft Azure, Amazon Web Services (AWS), or the Google Cloud platform.

  "Cloud-first" application development lends itself to Agile practices and a DevOps methodology in which all stakeholders (developers, operations teams, and quality assurance professionals) work together to deliver services quickly and accurately. Automation is central to "born in the cloud" application architecture.

- **Ever-evolving web applications:** Web applications became increasingly complex during the 2000s. The real issue is that web pages are more complex and rely on more code. For example, the average web page is now the size of an old-school DOS game (`https://mobiforge.com/research-analysis/the-web-is-doom`).

  Furthermore, applications are moving to the cloud and are increasingly application programming interface (API)–driven.

The result of these industry trends on the application security industry is a couple of growing challenges:

- **Incompleteness:** Application security scanners can't solve all the challenges alone. Scanners are important to include; the catch is that they have fundamental gaps in what they can find. Even though scanners bring great scale and consistency, they're faced with several technical challenges:

  - Scanners don't know how to make good choices about coverage (for example, which links might have risk associated with them and which links don't).
  - Scanners often report many false positives and false negatives.
  - Scanners fail when it comes to prioritization of the issues identified. They present the organization a long report with thousands of issues but no real way to distinguish what is important or not. Scanners can't calculate risk as it relates to the specifics of the target, its business purpose, or its data.

- Scanners don't understand the business logic of modern applications. As a result, application security scanners are not a complete solution, and today's most devastating attacks are still being identified by people.

- **Not enough people available to test applications:** Due to agile development methods, code is being deployed faster and faster. From a manual testing perspective, this is a new demand. Organizations need human-powered security testing. A company looking to engage people traditionally had two alternatives: Hire full-time employees or engage a consultancy. Unfortunately, the industry skill shortage has made it difficult to hire full-time staff and consultants can be quite costly.

Additionally, the traditional way for organizations to hire security talent is outdated. So, organizations are leveraging the flexibility of crowdsourced security platforms.

Today's requirements for an application security penetration test include the following:

- Cost that will enable higher frequency testing and greater coverage across an application portfolio
- Access to quality talent who can perform manual testing
- Strong integration with development processes in order to get issues fixed

In short, today's penetration tests must be agile, actionable, smart, and cost-effective.

What Is a Crowdsourced Pen Test?

A pen test targets an application to discover vulnerabilities, exploit them, and determine how well the application resists attacks.

A crowdsourced pen test is a pen test performed by freelance security researchers via a platform. This approach brings together the best elements from traditional pen testing per-

formed by consultancies with the global talent pool from bug bounty programs.

Three main elements distinguish a crowdsourced pen test from a traditional pen test that has historically been delivered by consulting firms:

- In a crowdsourced pen test, the talent is globally sourced.
- In a crowdsourced pen test, the pen test findings are delivered via an integrated platform.
- A crowdsourced pen test can be scheduled on-demand.

Chapter 2

How a Crowdsourced Pen Test Works

In This Chapter

- Understanding crowdsourcing in general
- Applying crowdsourcing to application security pen testing
- Knowing what to expect from a crowdsourced pen test

In this chapter, I explain what crowdsourced pen testing is, as well as the steps involved in actually carrying out such a test. I begin by defining *crowdsourcing*.

Crowdsourcing Defined

When you use a ride-sharing service such as Lyft, or schedule lodging with Airbnb, you leverage crowdsourcing. Today, crowdsourcing is becoming increasingly common and accepted as a normal way of undertaking professional or personal endeavors.

In a nutshell, *crowdsourcing* is obtaining input and/or services from a large number of people. It's a way to find people with special skills who are available to work and will be compensated for the work they do. The crowdsourcing model is also on-demand.

Your parents probably told you never to get into a stranger's car. Now, however, a French business named BlaBlaCar connects travelers for long-distance car rides. The company's primary factor used to match travelers for rides is how much they like to talk when they're on the road. Travelers can

choose from "bla," "bla bla," or "bla bla bla" and be matched up with someone with a similar conversation tolerance.

In a recent *Forbes* article, BlaBlaCar CEO Frédéric Mazzella said, "It's 10 million travels every quarter, which is about four times more than the Eurostar. A lot of people are using BlaBlaCar now to get around, and to help each other."

Crowdsourcing obviously offers several advantages involving mass intelligence to solve all sorts of different problems. Unlimited skill-set pools and a constant stream of new talent from around the world has gone through a rapid evolution. Reluctance based on fear of managing overhead and risky business, in general, is now slowly fading.

More and more companies within different industries are relying on crowdsourced services, and why should application security be the exception? Confidentiality? Trust? Technology professionals use these buzzwords to describe their concerns. However, these same people are using crowdsourced services for the majority of their other daily needs — for example, they use Lyft for transportation, order food from Forkable, redesign their houses through CoContest, and even recruit from Reflik. So, why not crowdsource their application security needs?

A simplistic process, constant education and awareness of developers, full workflow integration, as well as scalability and flexibility when it gets to remediation planning and re-assessing the risk posture are just a few of the advantages to crowdsourcing.

A Brief Pen Testing Overview

Organizations typically have a fixed budget to work with when designing a pen test strategy; they want to use the pen test to optimize both quality (talent, results) and coverage across an application portfolio or within a single application.

Penetration tests provide insight into an application's security by systematically reviewing its features and components. The pen test exercise improves coverage of application security

because the test explores the entire application instead of just focusing on one type of vulnerability or one particular section of code.

A vulnerability assessment focuses on enumerating known flaws and misconfigurations, such as those catalogued by the Common Vulnerabilities and Exposures (CVE) standard. Pen tests add context around the exploitability and impact of those vulnerabilities, perhaps identifying new ones along the way. The overall vulnerabilities provide a measure of risk associated with an app. The individual vulnerabilities highlight opportunities where better code and controls can reduce that risk.

Penetration tests follow trusted industry-standard methodologies that review topics like input validation, authentication, and access controls in order to identify flaws in the application's structure and implementation. Pen test results give developers confidence in their code and that their application protects its users, their data, and the systems upon which it is built.

Security testing after an application reaches production should never be the only stage at which security testing appears. Modern software development approaches like Agile and DevOps emphasize ever-evolving features, automated testing, and frequent releases. This accelerated development cadence makes it even more critical for security teams to keep pace.

What exactly is a crowdsourced pen test and what's different about it? Simply put, a crowdsourced pen test is like any pen test that reviews the security of its target; it just happens to be performed by a team from a crowd of qualified researchers rather than dedicated consultants (see Figure 2-1).

A crowdsourced pen test is a pen test performed by freelance security researchers via a platform. This approach brings together the best elements from traditional pen testing performed by consultancies with the global talent pool from bug bounty programs.

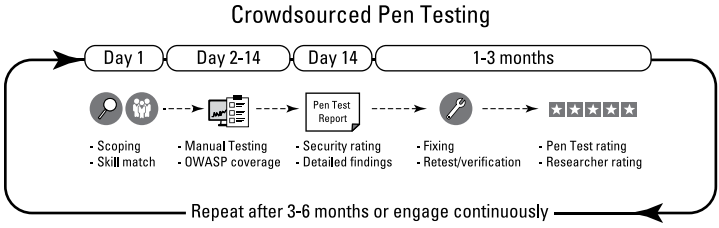


Figure 2-1: The crowdsourced pen test model helps organizations to find and fix security issues.

Getting Started with a Crowdsourced Pen Test

Conducting a crowdsourced pen test consists of completing the following four steps:

1. Match the right skills for a purpose-built team.
2. The team submits its findings.
3. Remediate issues.
4. Share the results.

These steps facilitate the overall vulnerability management process, from finding to fixing.

Step 1: Match the right skills for a purpose-built team

A crowdsourced pen test platform provides transparency into each researcher's individual skills, experience, and performance, so organizations can be matched with the researchers who will be the right fit for their specific security needs.

Just like you give your driver a rating when you complete a Lyft ride, and rate your Airbnb host when you stay at someone's home on vacation, crowdsourced pen testers are rated, too. A low rating results in a pen tester no longer being invited to projects, so the folks who are assigned to these projects are interested in developing and maintaining strong reputations based on all their project work. Pen testers are rated by their team members and by the clients they work for.

To kick off a crowdsourced pen test, an organization provides information about its application's technology stack and a few of the top researchers with matching skill sets are selected to do the project.

These folks work together as a team, exploring the complete application over a fixed time period. They work collaboratively to perform manual security testing related to topics like input validation, authentication, and access controls in order to identify flaws in the application's implementation.

Step 2: The team submits its findings

As the team members discover issues in the application, they submit reports to the organization through the crowdsourced pen test platform. The lead researcher is responsible for reviewing each report before it's submitted to ensure the report is valid. He or she also assigns a criticality rating to each report, based on likelihood and business impact.

Step 3: Remediate issues

The organization receives reports as soon as they're discovered and reviewed by the pen test lead. In some cases, receiving the report before the entire pen test is complete can give the organization extra time to get an important vulnerability fixed as soon as possible.

Fixing security issues is not just a technology problem; people and process are also required to get it done.

A crowdsourced pen test platform allows an organization to work with the findings and dynamically communicate with the pen testers for months after the initial test is complete. A platform may even integrate with developer bug tracking systems like JIRA or GitHub.

A platform enables collaboration between pen testers, the security team, and developers, since it centralizes data for your developers and can integrate with the tools that are already part of their workflow.

Step 4: Share the results

After the fixed time period is complete, the organization can also download a PDF summary report to share with internal and external stakeholders, such as development team leads or customers requiring proof of a technical security test.

From a timeline perspective, the crowdsourced pen test model differs from the traditional pen test model in a few different ways (see Figure 2-2):

- Scheduling a crowdsourced pen test is on-demand, whereas scheduling a traditional pen test with a consulting firm may require advanced notice.
- Reports in a crowdsourced pen test are delivered via the platform as the application is being assessed and issues are being discovered, whereas in a traditional pen test the results are delivered all at once, after the assessment has been completed.
- Validation, prioritization, and communication of results to development may be very manual and take some time in the traditional model. The crowdsourced pen test platform makes it easy by facilitating collaboration between security, pen testers, and development.
- Retest and verification may not be included in a traditional penetration test. Instead, remediated issues may get reviewed in the next penetration test. In the crowdsourced model, retest and verification are included for up to a year after the issues are first discovered, by the same pen test team that found them.

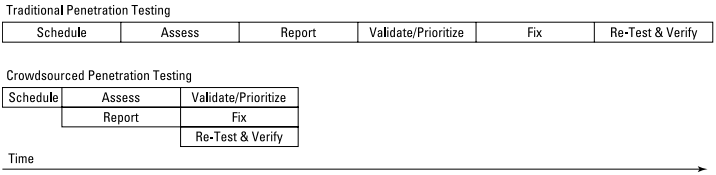


Figure 2-2: Comparison of timelines for traditional and crowdsourced pen testing.

Chapter 3

Collaboration at the Speed of DevSecOps

In This Chapter

- Reviewing the drivers for a new approach to application security pen testing
- Understanding how a platform enhances team collaboration

Today's development methodologies embrace continuous efforts. Continuous integration (CI) and continuous deployment (CD) processes have motivated fundamental changes to building and maintaining apps.

Introduction of agile methodology for most companies has proven to be extremely successful and has enabled improvements of speed, collaboration, and visibility across different teams and departments.

The ways in which development and operations teams interact is changing, and security must keep pace.

In this chapter, I discuss the practical collaboration and communication that must occur between teams in order to effectively find and fix application security issues. I also show you how a crowdsourced pen test platform facilitates these cross-functional interactions.

How Do Pen Test Findings Get Fixed?

Pen test findings provide a great feedback loop for developers to understand the real-world implications of writing insecure code. However, the relationship between the pen test team, security team, and development teams doesn't always run like clockwork.

Security organizations sometimes place heavy emphasis on defect discovery to find security issues, without sufficient focus on the processes and cross-functional relationships that are required to actually get those issues fixed.

Let's explore a common scenario: Say a company wants to integrate security into its SDLC, so it performs a penetration test or points some security tool at its application. The results are delivered in a PDF, and the security team downloads the report. Upon reviewing the report, the team members realize that there isn't that much that they themselves can do, and they need to engage development in order to get the issues fixed. So, they send a hasty email with an urgent subject line — something like, "We must fix everything now!" — and attach the PDF file without any further interpretation, prioritization, or guidance on what exactly to do with it. Every few days or weeks, they send a nagging email asking if the fixes have been implemented, without bothering to actually learn anything about how the development team does its work, or how its work is tracked, managed, and evaluated.

In this scenario, development teams don't have a good way to get their questions answered. The security team may put them in touch with the pen tester who originally found the issue or introduce the developers to a support team for the security tool, but that doesn't always happen.

So, how do you change the approach in order to improve collaboration and communication between security teams, pen testers, and developers?

Crowdsourced pen test platforms facilitate communication between security teams, pen testers, and developers by

providing a way for them all to interactively chat with each other. Getting all the relevant parties together in one place to talk about it can be very powerful. It can even make the difference between security issues getting fixed or not.

Because a crowdsourced pen test platform provides interactive support,

- Pen testers can ask developers about intended use cases for the application.
- Developers can ask pen testers questions about security findings.
- Pen testers can help developers understand exactly what needs to be done in order to remediate specific findings.
- Developers can ask security teams about each finding's criticality and how to prioritize fixes.
- When a finding is fixed, developers can ask pen testers to retest the issue and verify that the patch has been effective.

All this communication is necessary to ensure that pen test findings actually get fixed (see Figure 3-1).

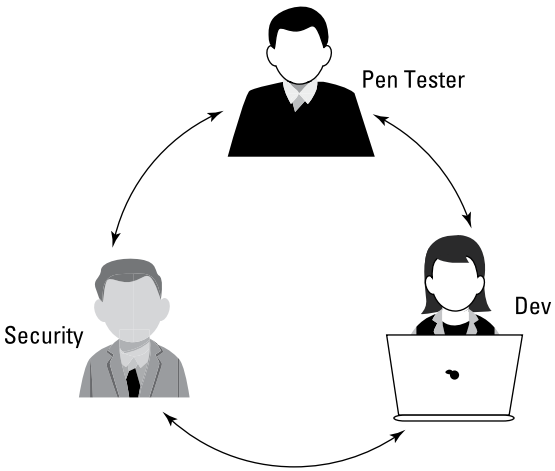


Figure 3-1: Security, dev, and pen tester roles must work together to address security issues.

Collaborating through a Platform

In previous decades, most manual application security workflows such as vulnerability disclosure or penetration testing were performed without the use of a platform to effectively manage the workflow and integrate with developer bug tracking systems. For example, consultancies would engage with clients by sending PDF reports in email, and they didn't leverage technology to ensure a consistent process and secure sharing of data.

In the 2000s, scanners brought both automation and consistency to the testing. Similarly, the crowdsourced application security platforms bring consistency to the manual processes and workflows. This makes the processes more smooth and accountable.

One major benefit of having a platform for delivering a service is that the centralization helps to aggregate and analyze data. This data can give insight into vulnerability trends, feedback on how well processes are working, and metrics on how well something is (or is not!) performing. In other words, you should approach this data with an eye toward using it to inform decisions or have actionable outcomes.

With the introduction of a crowdsourced security platform, metrics become the default and are seamlessly integrated. This makes application security more data driven and allows organizations to more easily benchmark and share their KPIs (see Figure 3-2).

From a talent perspective, the professional services industry has been lacking feedback loops in the form of a quality and review system. This changes with crowdsourced security platforms, because elaborate reputation systems now exist. Security professionals need to build and retain a strong, positive reputation to stay attractive on the platforms.

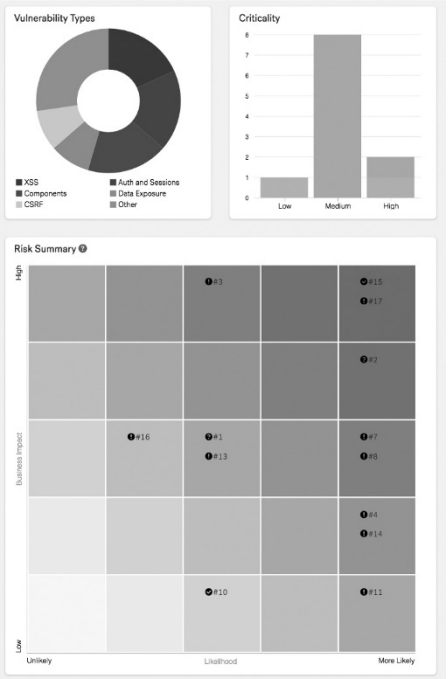


Figure 3-2: Security metrics are automatically calculated in a crowd-sourced pen test platform.

Using Pen Test Metrics to Improve Application Security

It's been said that "if you can't measure it, you can't manage it." It turns out that Peter Drucker never actually said that, but it is indisputable that measuring results and performance is crucial to an organization's effectiveness, and this definitely applies to application security.

A *metric* is a system or standard of measurement. A security metric measures activity to provide decision support for doing things better in the future. This data can help to answer questions that an executive or operator might have about pen test program attributes, using evidence-based information instead of opinion or anecdotes.

Limited resources are available to find, fix, and prevent software security vulnerabilities. Data and metrics are critically important to help practitioners make the best decisions about how to structure and measure the value of an application security program.

I've worked with a lot of organizations on metrics to show the value of their application security programs, and the challenge that comes up all the time is that organizations often don't have a single source of record for pen test findings, so they can't get the data to calculate their metrics. If the issues you're trying to fix are scattered around in PDFs and emails, it's going to be hard to count how many issues were found, let alone how many were properly addressed.

This data issue goes away completely in the crowdsourced pen test model, because the findings are delivered through the platform and the platform integrates directly with Jira, GitHub, whatever bug tracking system the organization is using to track and resolve all its software bugs — security or otherwise. The platform connects the dots between defect discovery and defect management, and findings are tracked from end to end. By building the process and data into the platform, metrics are no longer a heavy manual lift; they become the default.

Download the *Pen Test Metrics* e-book to dive deeper into this particular topic: `https://resource.cobalt.io/pentest-metrics-booklet`.

Chapter 4

Talent

In This Chapter

- Evaluating crowdsourced pen testers
- Using data and feedback loops to ensure high-quality talent

In this chapter, I discuss how trust and quality with crowdsourced pen testers is built and measured. With regards to talent, a crowdsourced pen test platform collects feedback on quality and performance, is transparent about the feedback, and provides direct access to the pen testers doing the work.

Evaluating Crowdsourced Pen Testers

Crowdsourced pen testers must be highly vetted and have the skills to perform detailed and high-quality penetration tests (see Figure 4-1). Thousands of people may want to become crowdsourced pen testers, but less than 5 percent of applicants are accepted. To even be considered, a security researcher must be recommended by someone who is already in the community. The vetting process also includes third-party government ID verification, social media account review, and a thorough interview over a video conference call. Anonymous testers are not allowed.

Skill matching is a critical component to achieving quality results in any penetration test. A typical crowdsourced pen test is led by a researcher who organizes the collaboration with two to three others. The lead will have extensive skill and experience within the security industry. The lead's role

includes validating the findings from the other researchers, coordinating with the client throughout the test phase, and communicating those findings in a final report.

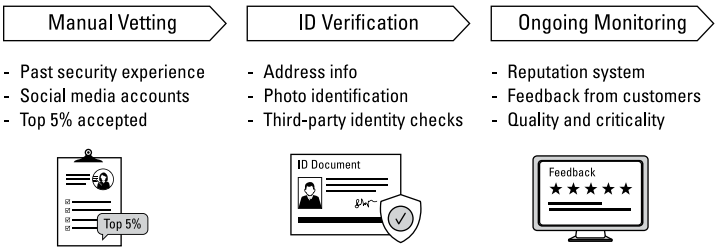


Figure 4-1: Vetting and verification of crowdsourced pen testers ensures that talent is skilled and provides quality results.

The researchers focus on testing the application for various security vulnerabilities. For web apps, this methodology aligns with the OWASP Top 10 and its Application Security Verification Standard (ASVS). Researchers may rely on various tools for analysis, but the majority of their effort is manual and serves as a complement to automated scanning. The researchers turn their understanding of the app into creative ways to bypass security controls or break assumptions in the app's design.

A pen test kicks off with a meeting between the lead researcher and the app's owners and developers. This discussion covers topics like verifying the scope of the testing, explaining key features and data flows, and ensuring test accounts are in place. It's also a chance to talk about some high-level threat models in order to help shape the pen test and make it more effective.

The test itself typically lasts two weeks. During this period the researchers distribute the work of reviewing the app's various features and components among themselves. They share notes with each other, describe tests they've tried or plan to try, and document vulnerabilities in the crowdsourced platform.

When the test is complete, the lead researcher collates the individual findings into a report that provides background about the engagement, as well as recommendations based

on themes or repeated issues that the researchers observed. For example, recommendations may note that a lack of input validation is pervasive throughout the application, that the application relied on trivially spoofed tokens for enforcing privilege levels, or that it's missing a centralized anti-CSRF tokenization.

The benefit of a pen test shouldn't just be in discovering vulnerabilities, but using that knowledge to reduce the risk associated with the application. Each finding has a risk score associated with it to help the application's developers understand and prioritize the work needed to resolve them. In this phase, the lead researcher is available to retest and verify that vulnerabilities have been fixed correctly. This step helps ensure that the fix addresses the underlying problem in the vulnerability.

The most important attributes of any penetration tester are skill set, experience, and performance. The best penetration testers also have strong communication skills and collaborate well with a cross-functional team.

Talented penetration testers are an essential component to achieving quality testing results. An experienced penetration tester knows how to do much more than run an automated scan (see Figure 4-2). He or she can think creatively (even maliciously) in order to mimic the attacker scenario against an application.

You want the penetration testers who are testing your applications to have skills that are matched to your application's technology stack. You want them to have many years of professional experience conducting security tests. And you want them to be highly rated by their team members and clients on their past performance.

Evaluating the quality or talent of a more traditional penetration tester can be difficult, but crowdsourced security platforms make it easy (see Figure 4-3). They provide this kind of information in a Hall of Fame and by displaying scores on penetration tester profiles.

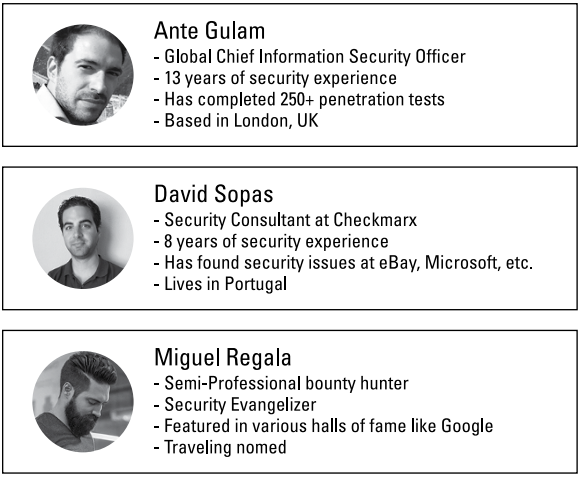


Figure 4-2: Quality security researchers have many years of experience and a track record of strong performance.

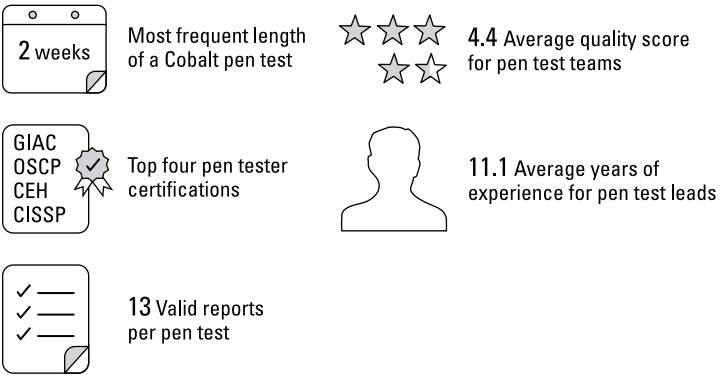


Figure 4-3: In a crowdsourced pen test platform, the talent data speaks for itself.

Here are a few statistics on the 2016 Cobalt crowdsourced pen testers:

- The average amount of professional experience for pen test leads is 11.1 years.
- The average quality score for ten test teams is 4.4 out of 5.
- The top pen test certifications are GIAC, OSCP, and CEH.

Sample Profile

Figure 4-4 shows a sample researcher profile. In the profile you can see the researcher's skills, performance ratings, halls of fame, and recent bug discoveries.

Figure 4-4: A sample researcher profile.

Chapter 5

Choosing the Right Security Testing for Your SDLC

In This Chapter

- Comparing vulnerability scanners to human-powered security testing
- Knowing which factors to consider when hiring a consultant
- Putting bug bounty programs in context

This chapter is about understanding the alternatives and how they may practically fit into a security program. Pen testing (crowdsourced or otherwise) isn't the only task; this chapter offers guidance on how best to use these alternatives.

Application Security Options

There are many application security controls, and none is inherently better than another, but security professionals should be aware of the various options available and pick the best one for their specific security requirements and business needs (see Figure 5-1).

Security scanners

Security scanners can be programmed to automatically identify vulnerabilities. A security scanner will never miss anything it is programmed to look for. At the same time, it will *always* miss everything it is *not* programmed to look for.

	Category Description
Security Scanners	- Programmable, with consistent and scalable results - Most powerful when customized - Use in conjunction with FTEs - Free to expensive
Security Consultants	- Human creativity - Locally sourced or travel paid - Highly educated and credentialed - Paid by the hour
Crowdsourced Bug Bounty	- Human creativity - Globally sourced - 1st to find a bug is paid a bounty - Public bug bounty has low signal-to-noise ratio
Crowdsourced Penetration Tests	- Human creativity - Globally sourced - Highly vetted, highly educated, and credentialed - Time-boxed and fixed price

Figure 5-1: Various application security options.

Vulnerability scanners produce a huge volume of data, which sounds good at first — until someone has to filter through the reports to determine what is valid and what is not. This usually requires an organization to have a full-time employee on staff who has a high level of expertise in the particular scanning technology in order to tune the scanner to get less noisy results.

Security scanning tools are most powerful when they're highly customized to a particular environment and application.

Security consultants

Quality security testing requires human creativity, and consultants provide testing as professional services.

Consultants are often locally sourced. If you want a particular consultant to be in a specific physical location, you have to pay for the consultant's travel expenses. These folks are in high demand, and they're busy on purpose. High billability is desirable because most of them will receive a company

paycheck and full-time equivalent (FTE) benefits whether they're working on a client project or sitting on the bench. Their projects may need to be scheduled in advance.

Sometimes, a traditional pen test consulting firm may provide a highly talented tester in the beginning (during the sales cycle and in a first engagement), but in some cases that person is later replaced by a junior person due to availability or other factors.

Security consultants are usually highly educated, credentialed, and expensive. They can be a great choice for security testing that requires in-person interaction with software, such as for embedded software or Internet of Things (IoT).

Crowdsourced bug bounty

A bug bounty program leverages a crowd of globally sourced researchers in competition to find security vulnerabilities in code.

In a public bug bounty, anyone in the world can submit a potential security vulnerability to an organization, and the first to find a valid bug will be paid a "bounty" and may also be featured in a Hall of Fame.

An organization running a public bug bounty pays the cost of each bounty and manages the overhead of reviewing and filtering all the reports (identification of false positives, duplicate removal, and so on). In a public bug bounty, only one out of ten incoming reports is likely to be a valid true positive.

Coordinated vulnerability disclosure is a model in which the security research community shares vulnerability information with the software owner before publicly disclosing it.

Crowdsourced penetration tests

A crowdsourced penetration test combines elements of bug bounty and traditional security consulting. Heavily vetted domain experts are selected from a crowd of globally sourced researchers and work collaboratively on a time-boxed (for example, two-week) penetration test of a web application, native mobile application, or set of APIs.

Crowdsourced penetration tests are a great starting point for an organization looking to jumpstart its application security program, agile shops that need frequent testing, and organizations looking for coverage across a large application portfolio.

Strategy: Three Factors to Consider

Every business will have a combination of different software environments and specific security needs. To determine which security testing methodology (or combination of methodologies) would be the best fit for a given scenario, the four categories described earlier can be evaluated using three key factors:

- Scalability
- Coverage
- Ease of use

Your specific testing goals should drive the prioritization of these factors.

Figure 5-2 summarizes the main points reviewed in this section.

	Scalability	Coverage	Ease of Use
Security Scanners	High	Low	Medium
Security Consultants	Low	High	High
Crowdsourced Bug Bounty	High	Low	Medium
Crowdsourced Penetration Tests	Medium	High	Medium

Figure 5-2: Table of scalability, coverage, and ease of use for various application security testing options.

Scalability

Scalability matters most for organizations that manage tens or hundreds of applications in their software portfolio. It matters less for an organization that has a single app.

Scanners

Scanners, like all technology tools, can scale very well for monolithic environments, assuming that all the requisite "care and feeding" is in place.

Consultants

Consultants don't scale as well, because they're premium experts who are paid by the hour.

Traditional pen tests still work fine — they just don't scale to budgets based on how many apps need testing.

Crowdsourced bug bounty

Crowdsourced bug bounty can attract many researchers if a program is appealing, but it doesn't necessarily guarantee that experienced, focused eyes will be testing the software. A robust vulnerability management process and sufficient analyst bandwidth to triage incoming reports is required to handle the low signal-to-noise ratio that is inherent in a bug bounty model.

A crowdsourced bug bounty scales great if you have more applications. You can just put all of them in scope. There is a hidden cost, though, in order to manage all the incoming reports.

Crowdsourced penetration testing

Crowdsourced penetration tests scale well due to their relatively low cost (compared to traditional security consultants) and high signal-to-noise ratio. Because a lead must review the findings before they're submitted and the researchers work collaboratively instead of competitively, only high-quality findings are delivered to the organization.

Coverage

Malicious attackers will try everything they can to reach their targets. In order to mimic that in security testing, it's important to have coverage across an application portfolio and through comprehensive test cases.

Scanners

Scanners can't think creatively or find design flaws; they only look for what they're programmed to find. Scanners have predictable coverage. The areas they don't cover are also predictable.

Consultants, bug bounty hunters, and crowdsourced penetration testers can think creatively and brainstorm misuse and abuse cases, which a scanner cannot do. They can consider application business logic and identify design flaws in addition to "just" finding bugs.

Scanners will only cover known vulnerabilities. They have difficulty understanding business logic.

Consultants and crowdsourced penetration testing

Consultants and crowdsourced penetration testers often have a procedural approach to ensuring coverage — a checklist of sorts — that includes the OWASP Top 10 or the ASVS. Bug bounty programs are continuous and in theory have an "infinite" number of eyeballs on the problem, but the approach is more "scattered" and there is no guarantee that a particular bug bounty program will attract technology-specific skills or a large volume of researchers.

Consultants onsite can cover white-box testing and follow a methodology.

Crowdsourced bug bounty and penetration testing

Crowdsourced bug bounty and penetration testing both have the advantage of a globally sourced pool of researchers. Bug bounty provides a potentially larger volume with a broad spectrum of skill sets and experience; crowdsourced penetration tests include multiple researchers who are highly vetted, skilled, and focused.

In a crowdsourced bug bounty, you never know if someone actually looked at "application number 14" (as an example) and checked the mobile API for authentication issues.

There is a procedural element to the pen testing approach, which ensures higher consistency and more coverage than bug bounty. Also, because real humans are involved, there is coverage across attack scenarios that scanners might not understand.

Ease of use

Ease of use matters. This section looks at the total cost of ownership (people, process, and technology) for each of these security testing options.

Scanners

Scanners need to be customized in order to get the most value. Scanners require manual tuning in order to effectively crawl web applications and scenario-specific manual configuration to test any sort of business logic. Certain types of vulnerabilities (authorization and session management) can be particularly difficult for scanners to find.

It's easy to point a scanner to a public facing website. It's more difficult to point a scanner to authenticated business workflow where ten or more steps are required to test it.

Public bug bounty programs

Scanners and public bug bounty programs both generate a lot of findings (low signal-to-noise ratio) that must be filtered manually in order to get to the set of true positive findings. Scanners produce a large number of false positives unless they are carefully tuned and results are filtered, and the bug bounty model produces many duplicates.

Starting a bug bounty is easy. Any developer can just get started without too much budget or setup cost. It's difficult, though, to run and scale it. How do you manage 100+ bounty hunters of different quality? How do you decide on bounties and manage escalations?

Bug bounties have weak filter effects. They produce noisy findings that require manual overhead to review.

Regardless of whether this task falls on the organization sponsoring the bounty program or the platform operating the program, someone will be responsible for identifying which findings require follow-ups and fixes, and which findings can be forgotten.

Pen tests tend to function as a band-pass filter — they overwhelmingly produce valid findings (bugs to be fixed) and attenuate noise that would be distracting to developers.

Consultants

Consultants deliver fewer findings with a higher signal-to-noise ratio, typically in a PDF report. An organization might manually enter the details of the findings into a bug ticketing system so that the information gets to a developer who can remediate the issues. Or, the findings remain isolated in a PDF on a SharePoint site or spread out in various email threads.

With security consultants, you just call them and they show up. They know what to do without too much explanation.

Crowdsourced penetration testing

Crowdsourced penetration testing provides higher-quality findings because a purpose-built team works together, eliminating the creation of many duplicate and false-positive findings. This option also provides support to an organization past the "find" phase and throughout the "fix" phase by allowing an organization to communicate directly with researchers and request retesting and verification of patches.

There is a setup cost to get started because of the high-quality talent. However, once you're up and running, the freelancers know what to look for and don't need to much hand-holding or maintenance.

With a crowdsourced pen test platform you can get more transparency into who's doing the work and that person's history of work (for example, rating of his previous findings and feedback from previous pen test customers). The crowdsourced platform can show ratings over time and has rating feedback loops built in.

What Matters Most: Find and Fix

Whatever combination of people, process, and tools you choose, you want to find as many true positive findings as possible so they can be addressed.

The reality is that security bugs and flaws exist in your software, regardless of whether you know they're there or not. But you can't fix security issues if you can't find them.

When you've performed defect discovery in order to find as many true positives as possible, the next step — by no means a trivial one — is to communicate them to the developer team, get them to prioritize the fixes, get them to remediate the issues, and ideally prevent the same issues from coming up again. Fixing security issues is not a technology problem; people and process are also required to get it done.

You've got to find the issues — the real ones — and you've got to fix them. All while managing cost and coverage across an application portfolio.

At the end of the day, you want to take an honest look at the application security testing options available and evaluate them based on the factors that matter the most to you and your organization.

Consider your organization's security objectives and choose accordingly:

- If your goal is to build DevSecOps into a large development team, consider using security scanners as a foundation for your testing program and augment the scans with crowdsourced penetration testing.
- If your organization seeks SOC 2 compliance, consider working with consultants in order to ensure coverage and a strong brand.
- If you want to establish a public communication channel with external security researchers, consider employing a public bug bounty program.

- If you want to increase the frequency of penetration tests on your application portfolio and integrate security testing with your release cycle, consider crowdsourced penetration testing.

The choice is yours! Make an informed decision, not simply one out of habit.

Chapter 6

Ten Useful Sources for Information on Crowdsourced Pen Testing

This book outlines the case for crowdsourced penetration testing, but you may want a little more. The following resources will help as you perform your research and conduct due diligence:

- **Crowdsourced Application Security: The Human Power** (`www.itspmagazine.com/from-the-newsroom/crowdsourced-application-security-the-human-power`): This Experts Corner article by ITSPmagazine describes Caroline Wong's session at the 2017 OWASP AppSecCalifornia conference on crowdsourced security. The article describes application security options and pointers on how to evaluate each according to the criteria: scalability, coverage, and ease of use. The slides from Caroline's presentation can be found at `https://appseccali2017.sched.com/event/8aEV/crowdsourced-security-the-good-the-bad-and-the-ugly`.
- **Modern Pen Testing** (`https://youtu.be/3TW2-zc47B8`): This in-depth recording features Sven Schluter, the Head of Assurance, Germany for Context Information Security. Sven explains how penetration testing has changed to suit new innovations in technology. He conducts a detailed investigation into the different worlds of bug bounties, research, and penetration testing.

- **Deconstructing and Rewiring Bug Bounty Programs** (https://blog.cobalt.io/deconstructing-and-rewiring-bug-bounty-programs-f76c2b72dd11#.l1ds1xv14): This extensive blog post by Jacob Hansen describes lessons learned from more than 200 bug bounty programs, particularly the significant management costs involved. Jacob also outlines the return on investment (ROI) for other crowdsourced security services, such as vulnerability assessments and penetration tests.
- **Integrating Crowdsourced Security into the Agile SDLC** (www.slideshare.net/FrancoisRaynaud/devseccon-asia-2017-ante-gulam-integrating-crowdsourced-security-into-agile-sdlc-72604779): These slides were presented by CISO Ante Gulam at DevSecCon Asia 2017. In this presentation, Ante discusses why the Agile SDLC needs agile security and talks about how transparency builds trust. If you enjoy Ante's direct and insightful style, check out his popular presentation, "Building Resilience into Information Security" (www.slideshare.net/ante0303).
- **A Scrutiny of Crowds** (https://blog.cobalt.io/a-scrutiny-of-crowds-penetration-testing-with-cobalt-a062d749dfba#.vtvaa4n23): Mike Shema is the brains behind the respected online resource, Deadliest Web Attacks (www.deadliestwebattacks.com). In this blog post, he muses on today's development ecosystem and the combination of the best elements of the bug bounty model and traditional penetration testing.
- **Crowdsourced Penetration Testing** (https://youtu.be/rpCMpFzSasc): In this video, Mike Hendrickson, Vice President for Content Strategy at O'Reilly Media, Inc., interviews Jacob Hansen and Caroline Wong at the inaugural O'Reilly Security Conference in New York City about crowdsourced penetration testing.
- **Bug Bounty Ethics and the Ubering of Pentesting** (www.danielmiessler.com/blog/bug-bounty-ethics-uber-pentesting/#gs.sbQqDfw): In this blog post, Daniel Miessler takes a closer look at some of the controversial comparisons that the pen testing community has made about bug bounty programs to Uber and similar ride-sharing services. Are bug bounties exploitative of bounty hunters? Read Daniel's blog for an informed perspective.

- **AppSec Disrupted and AppSec Reanimated** (`https://webinar.cobalt.io`): These webinars cover topics such as crowdsourced pen testing, software development life cycle fails, and preparing for modern AppSec threats.
- **Bad Medicine: Contradictions of Bug Bounty Programs** (`https://youtu.be/CgPsZA4ORJA`): In this video, LinkedIn's CISO Cory Scott talks about "peak bug bounty hype" and the side effects of adopting a bug bounty program.
- **Pen Test Metrics, by Caroline Wong and Mike Shema** (`https://resource.cobalt.io/pentest-metrics-booklet`): This downloadable e-book describes the key metrics that are required to demonstrate value and ROI of any modern pen testing program. It includes 2016 data from a crowdsourced pen test platform and lessons learned from hundreds of pen test programs.

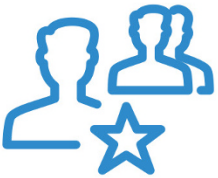
Crowdsourced Pen Testing

Delivering Application Security at Speed

Cobalt's agile SaaS platform delivers actionable results that empower agile teams to pinpoint, track and remediate software vulnerabilities.

Collaborative Team

1 Certified Lead plus 1-3 technical domain experts



Actionable Data

Smart, secure, and collaborative SaaS platform

Creative Approach

Access to our globally sourced talent pool

Vuln Prioritization

Pen Test Report incl. findings & executive summary

Visit us at cobalt.io for a demo today

Superior pen test results without generic scanners and overpriced consultants

Web applications are becoming increasingly complex, applications are moving to the cloud, and code is being deployed faster and faster. That's why traditional ways of performing application security pen testing have changed. We're now in a world where application security pen tests can be scheduled on-demand and findings can be tracked — from discovery through remediation, retest, and verification — centrally in a SaaS platform. This book helps you understand the industry trends that led to the emergence of crowdsourced security and how it can be applied to application security pen testing.

- ***Reap the benefits of a highly vetted pool of global security researchers — nothing beats human creativity when it comes to identifying vulnerabilities and business logic flaws in applications***
- ***Dodge the hidden cost of public bug bounties — get focused testing that explores the complete application using a purpose-built team matched to your application's specific technology stack***
- ***Embrace the collaborative platform — work with pen test findings and dynamically communicate with pen testers for months after the initial test is complete***
- ***Prepare to transform the remediation model — integrate platform workflows with developer bug tracking systems and expect retest and patch verification for each finding by the pen test team that uncovered the issue in the first place***

Open the book and find:

- **A breakdown of critical trends that are shaping application security today**
- **The three most common pitfalls of traditional pen testing**
- **Sample profiles of real-life crowdsourced pen testers**
- **Tips on how to integrate application security testing with your organization's development processes**

Go to Dummies.com® for more!

FOR DUMMIES®
A Wiley Brand

ISBN: 978-1-119-41607-4
Not for resale

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.