Table of Contents

About the Report

Here at Cobalt, we've done over 350 penetration tests to date. The information included in this report (Time to Fix, Vulnerability Types, Findings Criticality, Issues Fixed) is summary data from all of the penetration tests performed in 2017.

Additionally, we provide survey data (Portfolio Coverage, Pen Test Frequency) from 75 respondents in security, management, operations, DevOps, product, and developer roles.

All data has been anonymized to protect the privacy of our contributors.

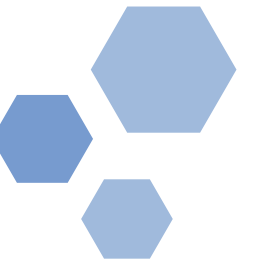
Team

Authors

Caroline Wong
Mike Shema

Production

Chris Tilton
Esben Friis-Jensen
Julie Kuhrt



Cobalt
Pen Testing as a Service

What is your current confidence level in your application security program?

Are you tracking these metrics?
Maybe you should.

Executive Summary

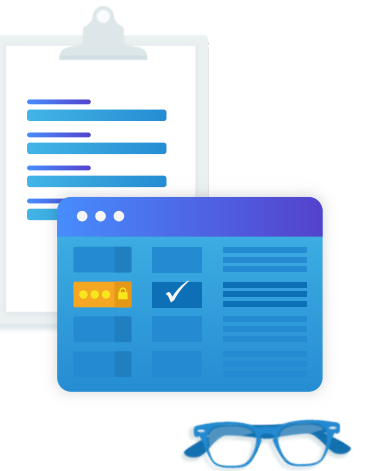
AppSec pen tests are time-limited, fixed price, manual application security tests performed by top security researchers.

Organizations typically have a fixed cost to work with when designing a pen test strategy, and want to use it to optimize quality (talent, results) and coverage (across an application portfolio and within an application).

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.

This report details several application security metrics used to measure the effectiveness of penetration testing at both program and engagement levels.



Program Level Metrics

Portfolio Coverage

An organization should apply security controls in a risk-based manner across its entire application portfolio.

Pen Test Frequency

An organization should conduct a penetration test on critical applications once a quarter.

Time to Fix

Critical findings should be fixed as soon as possible.

Engagement Level Metrics

Vulnerability Types

How real world vulnerabilities map to common references like the OWASP Top 10 categories.

Findings Criticality

Some findings are more critical than others.

Issues Fixed

Finding is great, but fixing is what actually improves application security.

Introduction

What is an application security penetration test?

AppSec pen tests are time-limited, fixed price, manual application security tests performed by top security researchers. Organizations typically have a fixed cost to work with when designing a pen test strategy, and want to use it to optimize quality (talent, results) and coverage (across an application portfolio and within an application).

Penetration tests provide insight into an application's security by systematically reviewing its features and components. This type of exercise improves coverage of an application's security because the test is intended to explore the complete application rather than just focus on one type of vulnerability or one particular section of code. Penetration tests follow methodologies related to topics like input validation, authentication, and access controls in order to identify flaws in the application's implementation. The results of these tests help give developers a sense of confidence in how well the application protects its users, their data, and the systems it's built upon.

Conducting security testing after an application reaches production should never be the only stage where security appears in an application's lifecycle, but it's still an important spot for it. Modern software development approaches like agile and devops concepts emphasize frequent releases with ever-evolving features. This rate of development makes it even more critical for security teams to keep pace with releases.

What is a security metric?

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 a particular area, such as penetration testing, using evidence-based information instead of opinion or anecdotes.

Best practices in application change and go in and out of date very quickly. In application security, one size doesn't fit all. Standards and controls are built based on years of practical security experience in real organizations - but this is something which is constantly changing.

Today, every application security practitioner needs to know how to optimize his or her unique program using metrics.

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

Here are a few benefits of using metrics to evaluate an application security program:

1. Measurement provides visibility.

2. Measurement educates and provides a common language for understanding a security program.

3. Measurement improves. It enables the best possible management of the security program, it enables

investment planning and decision making, and it drives necessary change throughout the organization.

When designing metrics for application security, you want to specify an objective and assess security activity against that objective. This will allow you to determine how much progress you are making in terms of achieving your goal.

Measurement should be conducted frequently and consistently in order to generate trend data. Once you begin to gather data and calculate metrics, that information can be visualized in a chart or graph.

Visualizations can make the data easier to consume and analyze. If the goal has been clearly stated and the visualization has appropriate labels, then you should be able to easily see from the visualization what is going well and what needs improvement.

Types of application security metrics for penetration testing:

In this report, we discuss two groups of application security metrics.

Program Level Metrics

The first group can be used to analyze the overall penetration testing setup for an organization and make strategic decisions.

Engagement Level Metrics

The second group can be tied to an individual penetration testing engagement to determine how well it is performing.

Program Level Metrics

Portfolio Coverage: An organization should apply security controls in a risk-based manner across its entire application portfolio.[1]

When it comes to penetration testing, the term "coverage" can mean a variety of different things. This same word may be used to represent coverage across a software portfolio (how many applications have been tested out of a total number of applications), indicate exploration within a single application (using a checklist approach to document what has been tested, such as the OWASP Top 10 or ASVS), or specify how much data about the application has been shared with the tester (white box, grey box, black box). It's always important to clarify the specific context that you're talking about.

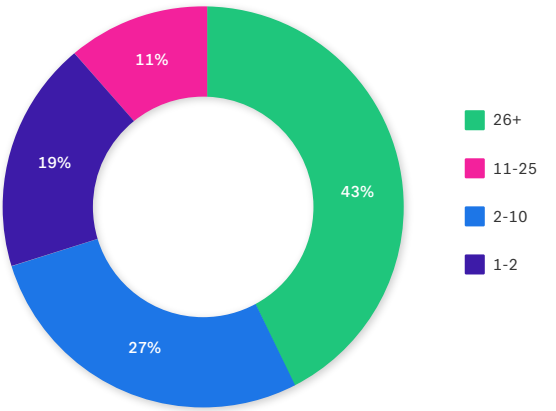
Here, we are talking about coverage across an application portfolio. Ideally, an organization should apply security controls in a risk-based manner across its entire software portfolio. That might mean performing 3rd party penetration tests on the critical applications and running a scanner on the rest. Due to limited resources for application security testing and a software portfolio consisting of applications of varying risk levels, application security controls are not one-size-fits all.

Let's say an organization has decided to conduct penetration testing on all critical customer-facing applications. If that organization has ten critical customer-facing applications in its software portfolio, then its goal should be to pen test all ten of the applications.

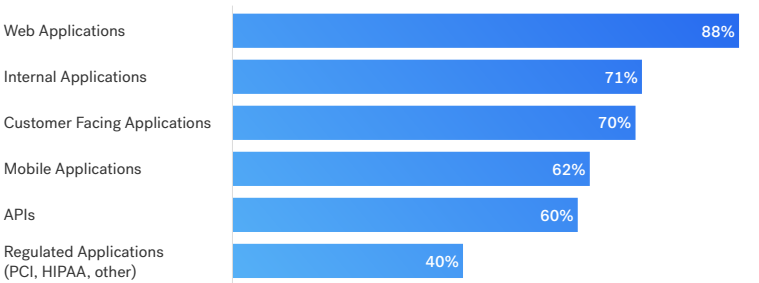
[1] Data based on survey responses

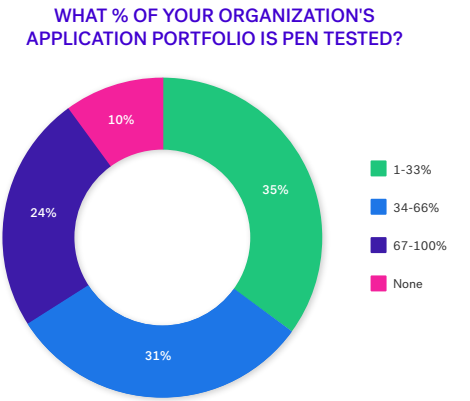
Goal	An organization should conduct a penetration test on every critical web application, mobile application, and API in its software portfolio.
Question	What percentage of critical applications has been penetration tested in the last 12 months?
Metric	% = # critical applications tested / total # critical applications in portfolio

NUMBER OF APPLICATIONS



TYPES OF APPLICATIONS





Pen Test Frequency: An organization should conduct a penetration test on critical applications once a quarter.[2]

Many regulations, such as PCI DSS, SOX, and HIPAA require an annual penetration test from a third party. We see the annual compliance requirement as the bare minimum that an organization should do for critical applications. Security savvy organizations know that as time goes on and code changes (and/or requires patches to stay up to date), that semi-annual or quarterly penetration tests are a much better idea.

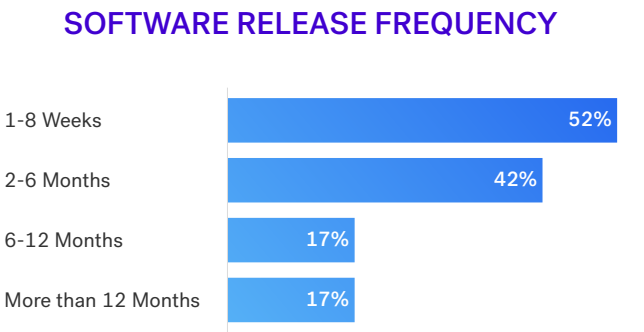
Attackers are constantly evolving the way that they attempt to breach applications. In order to stay one step ahead, penetration testing should be conducted periodically on an application, even if no recent changes have been made. New vulnerabilities are discovered all the time, and an application may be vulnerable to attack if software updates have not been installed.

2 Data based on survey responses

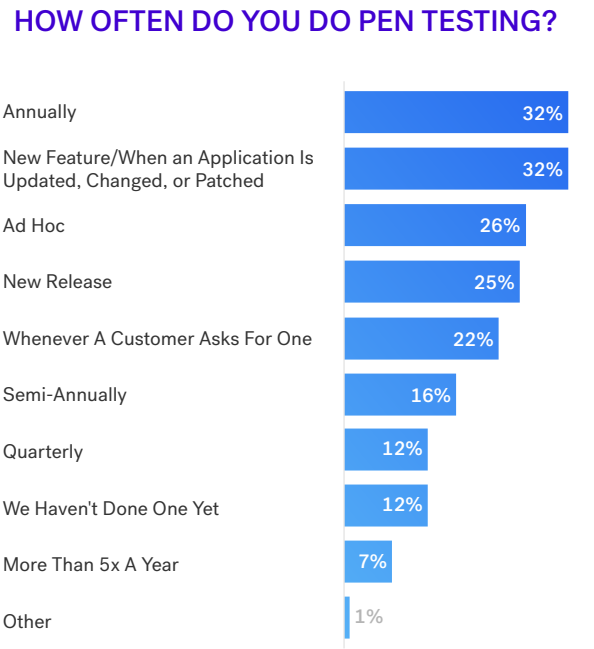
Organizations often ask, am I testing frequently enough compared to my peers?

It's also important to ask, am I testing enough compared to my development cycle? For example, if I'm releasing new versions of my critical apps six times a year but only doing penetration testing twice a year, then should I consider more frequent testing? Yes.

Goal	An organization should conduct a penetration test on critical applications once a quarter.
Question	How many critical applications were pen tested this quarter?
Metric	# critical apps that have been pen tested this quarter / # total critical apps



Pen Test Frequency: An organization should conduct a penetration test on critical applications once a quarter. [2]



[2] Data based on survey responses

Time to Fix: Critical findings should be fixed as soon as possible.[3]

Finding is important, but fixing is what actually matters. It's easy to say that we want to fix all the issues that have been found, however it's not so easy to make it happen. Developers are focused on developing new features and meeting deadlines, and have limited bandwidth to remediate security issues. It's certainly not possible to fix all the security issues at once. They have to be prioritized and addressed over time.

What kinds of metrics can be used to help with prioritizing which issues should get fixed and when?

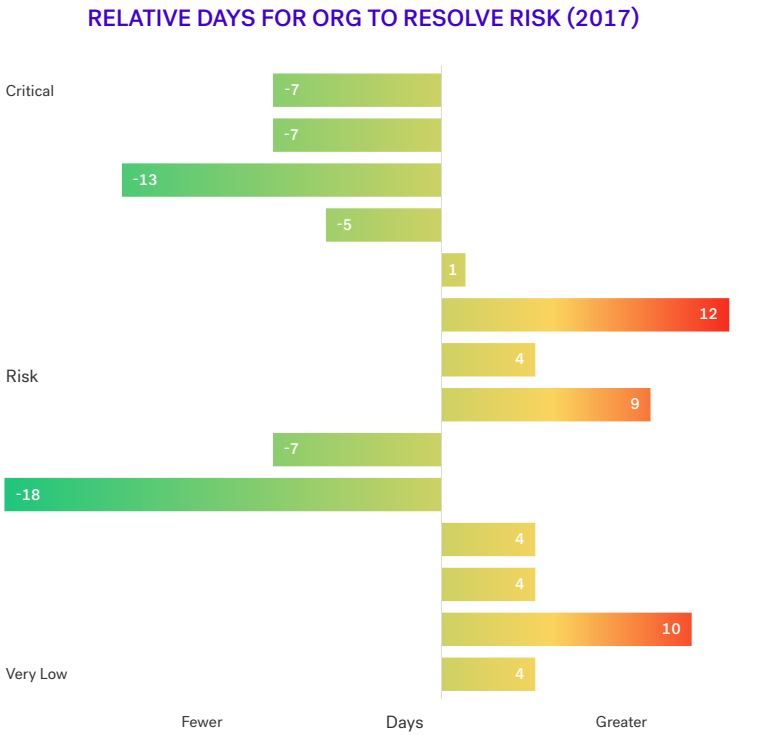
Many organizations require that findings be fixed within a certain period of time, depending on the criticality of the finding. For example, an ecommerce business might require that critical findings discovered on its customer facing applications be fixed within 48 hours, high severity findings be fixed within 10 days, medium severity findings be fixed within 30 days, and low severity findings be fixed within 90 days.

Talk with your developers in order to understand what a reasonable set of Service Level Agreements (SLA) is for your organization. It's a good idea to have a specific, documented SLA and to track against it. Over time, you can use historical trend data to determine if it's an appropriate SLA or if it needs to be adjusted. Mature organizations should enforce a consequence when SLA requirements are not met.

[3] Data from Cobalt's pen testing as a service platform, based on 250+ pen tests conducted in 2017

Time to Fix: Critical findings should be fixed as soon as possible.[4]

Goal	Fix critical findings as soon as possible.
Question	What's the average time-to-fix for critical pen test findings?
Metric	Average (time-to-fix for critical findings)



[4] Data from Cobalt's pen testing as a service platform, based on 250+ pen tests conducted in 2017

Engagement Level Metrics

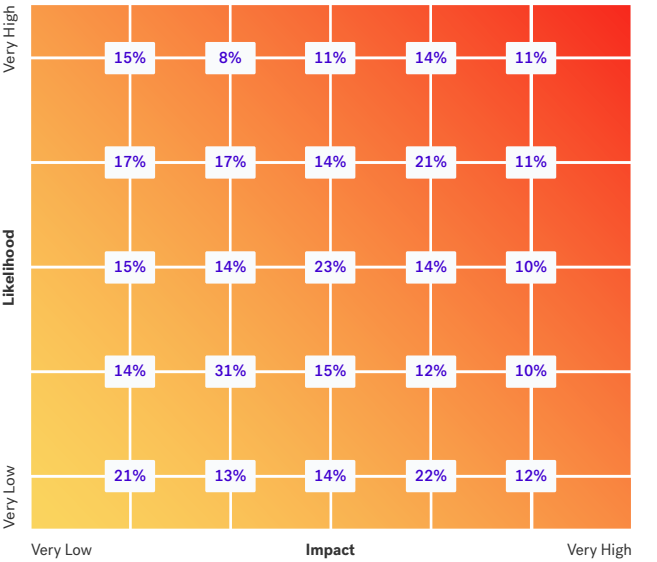
Findings Criticality: Some findings are more critical than others.[5]

The criticality of a penetration test finding can be calculated by considering the potential impact to the business as well as the likelihood of occurrence. Higher criticality security findings should be remediated before lower criticality security findings, especially those that might be easily exploited.

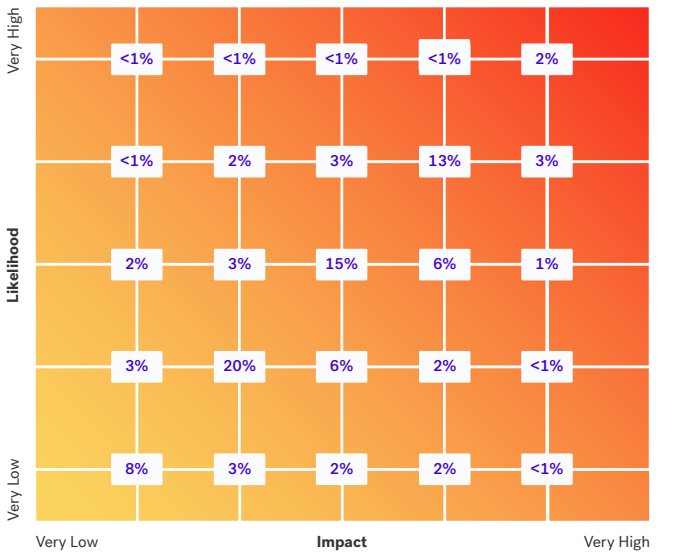
Goal	Prioritize remediation of penetration test findings
Question	How many high criticality findings were found in the last penetration test? How many medium criticality findings? How many low criticality findings?
Metric	Count the number of penetration test findings at each level of criticality (low, medium, high)

5 Data from Cobalt's pen testing as a service platform, based on 250+ pen tests conducted in 2017

CHANCE OF A FINDING PER PEN TEST (2017)



DISTRIBUTION OF ALL FINDINGS (2017)



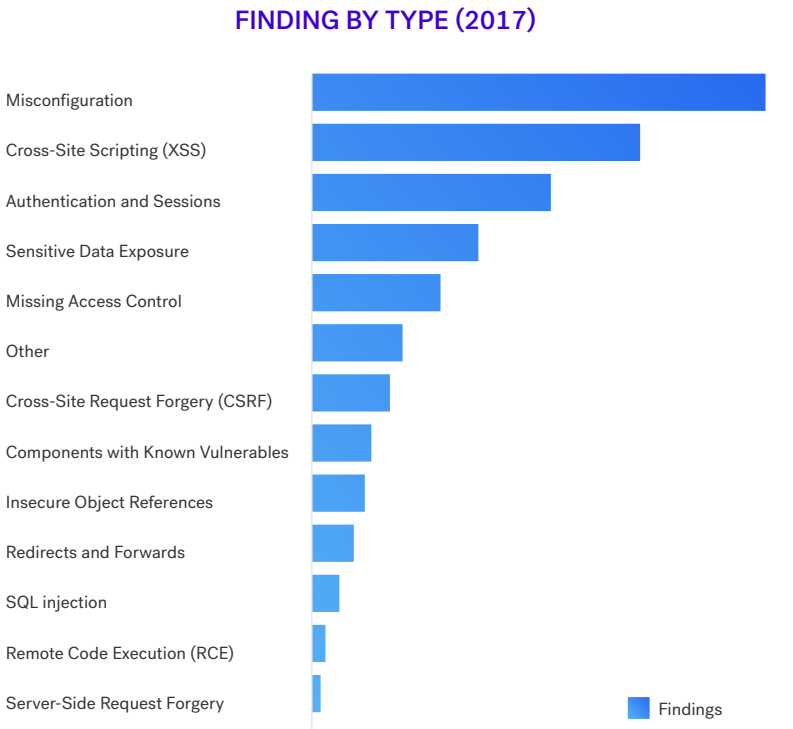
Vulnerability Types: How real world vulnerabilities map to common references like the OWASP Top 10 categories.

By visualizing and analyzing how many instances of each vulnerability type have been found in a penetration test, an organization can begin to strategically eliminate certain types of vulnerabilities by focusing prevention strategies on a particular vulnerability type.

The OWASP Top 10 contains a list of common web applications security risks, however each organization will have its own unique "Top 10" list. **If you know what yours is, you can and should use this information to eliminate entire categories of security vulnerabilities** by putting into place focused developer training, writing custom static code analysis rules, integrating tests for these types of security vulnerabilities into QA testing, etc.

Goal	Prioritize remediation of security defects
Question	What types of security vulnerabilities were found in the most recent penetration test? What's the category with the greatest number of instances found? What's the category with the next greatest number of instances found?
Metric	Count the number of security defects of each vulnerability type

Vulnerability Types: How real world vulnerabilities map to common references like the OWASP Top 10 categories.[6]



6 Data from Cobalt's pen testing as a service platform, based on 250+ pen tests conducted in 2017

Issues Fixed: Finding is great, but fixing is what actually improves application security.

An organization should track how many issues found in each penetration test got fixed. Once you've performed penetration testing in order to find as many issues as possible, the next step - by no means a trivial one - is to communicate them to the development team. The development team is a critical stakeholder when it comes to prioritizing the fixes, remediating the issues, and ideally preventing the same issues from coming up again.

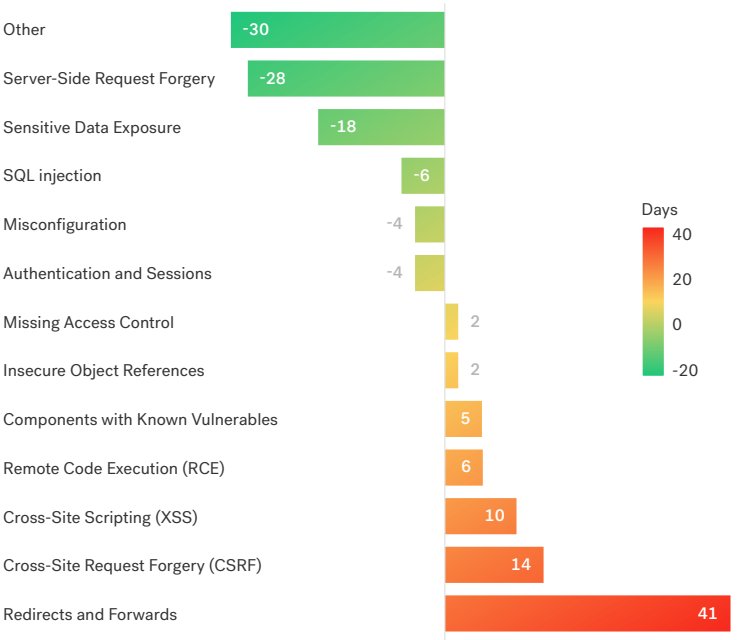
Fixing security issues is not only a technology problem; people and process are also required to get it done.

It's important to measure the effectiveness of a penetration testing capability - and by effectiveness, what I mean is how the penetration testing results actually improve the security of the application code. How many of the found security issues have actually been fixed?

Goal	All valid security defects should be fixed.
Question	How many found security defects have been fixed?
Metric	# security defects fixed / # security defects found

Issues Fixed: Finding is great, but fixing is what actually improves application security.[7]

RELATIVE DAYS FOR ORG TO RESOLVE A VULN TYPE (2017)



7 Data from Cobalt's pen testing as a service platform, based on 250+ pen tests conducted in 2017

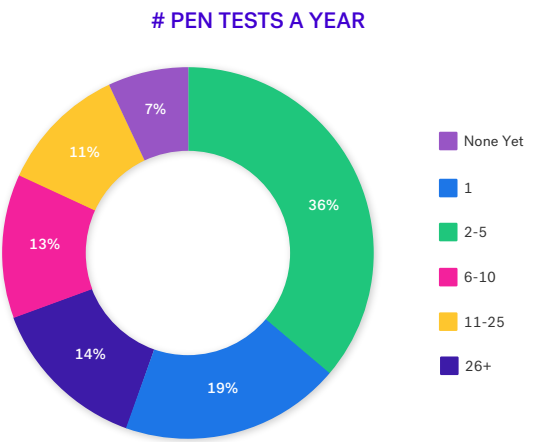
Survey Data

In 2017, we collected data from 75 survey respondents in security, management, operations, DevOps, product, and developer roles.

These respondents work in a variety of different industries, including Cloud/SaaS, Finance, Retail/eCommerce, Healthcare, and others. Their organizations use a variety of different approaches to software development: 60% of respondents do Agile/DevOps and 28% do Waterfall; 58% builds software internally and 49% works with third parties to develop their software.

In addition to the data provided in Portfolio Coverage and Pen Test Frequency above, here's what survey respondents had to say about why they do pen testing, what's most challenging about pen testing applications, which metrics they use, and how they manage a pen testing budget.

How many pen tests do you do a year?

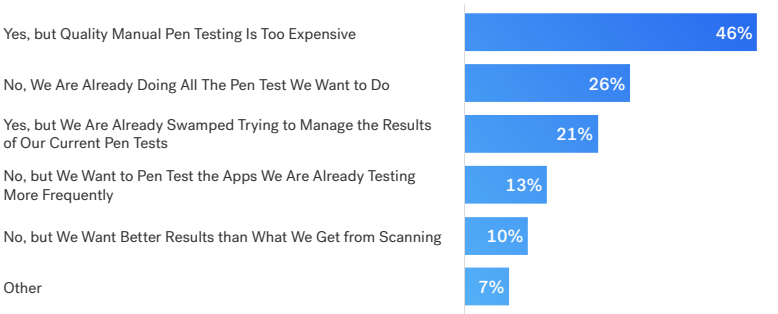


Drivers for penetration testing include improving application security, adhering to a software development lifecycle, validating that security issues have been fixed, and providing a risk assessment for third party vendors.

Pen testing challenges include finding the right people, ineffective or noisy tools, and cost.

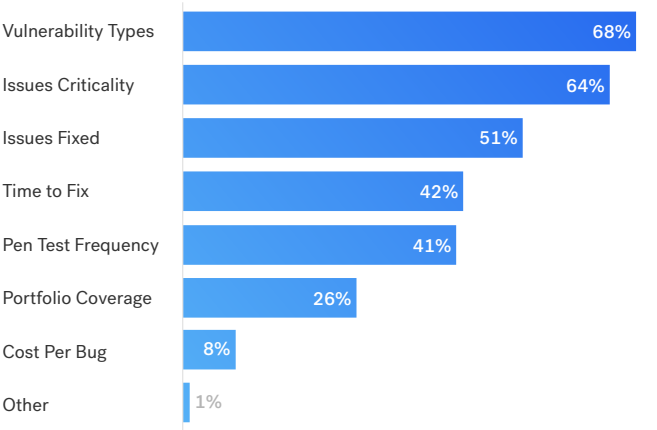
WHAT IS MOST CHALLENGING ABOUT PEN TESTING APPLICATIONS?

DO YOU WANT TO PEN TEST MORE APPLICATIONS?



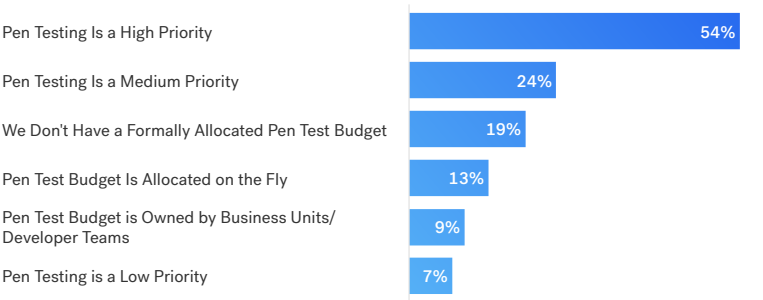
Common security metrics in use include vulnerability types, issues criticality, and issues fixed.

WHICH OF THESE METRICS ARE YOU ACTIVELY USING?



When it comes to prioritizing a pen test budget, there's a lot of variability. For many organizations it's a high priority, but others don't have a formally allocated budget for pen testing.

WHEN YOU'RE ALLOCATING YOUR ANNUAL INFORMATION SECURITY BUDGET, HOW DO YOU PRIORITIZE PEN TESTING?



Conclusion

At Cobalt, we believe in people and data. The most interesting and important security findings cannot be discovered via automated means alone. Human intelligence and creativity is necessary. Data sharing between pen test teams, security teams, and development teams is critical to improving the security of applications. We hope that the information contained in this report is useful to you and your organization.

Want to discuss this report? Let's talk. Contact us at hello@cobalt.io

About the Authors

Caroline Wong is the Vice President of Security Strategy at Cobalt

Her close and practical information security knowledge stems from broad experience as a Cigital consultant, a Symantec product manager, and day-to-day leadership roles at eBay and Zynga.

Caroline authored the popular textbook Security Metrics: A Beginner's Guide, published by McGraw-Hill in 2011. She was featured as an Influencer in the 2017 Women in IT Security issue of SC Magazine and has been named one of the Top Women in Cloud by CloudNOW.

Mike Shema is the Vice President of SecOps and Research at Cobalt

His experience with information security includes managing product teams, building web application scanners, and consulting across a range of infosec topics.

Mike has put this experience into books like the Anti-Hacker Tool Kit and Hacking Web Apps. He has taught hacking classes and presented research at conferences around the world.