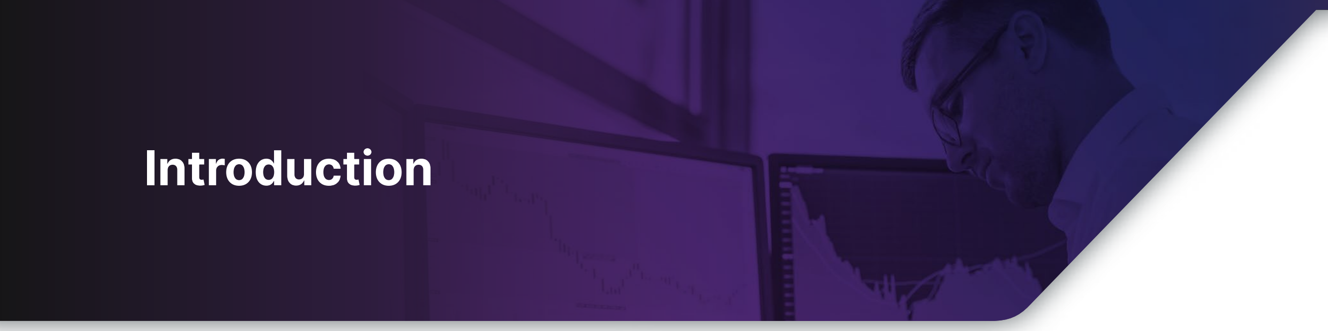
EBOOK

BISHOPFOX

FORTIFYING YOUR APPLICATIONS

A Guide to Penetration Testing

Table of Contents



Introduction

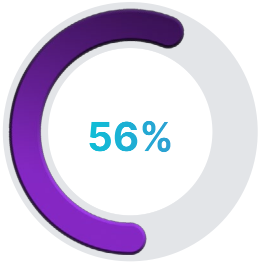
According to recent **research,** web application attacks were the leading cause of data breaches for six out of the last eight years. Application security is and should be a top concern for developers and security professionals alike.

Whether you have conducted dozens of application penetration tests before or are about to start your first, this eBook offers guidance we hope you find valuable. We'll explore key aspects of application penetration testing, questions to ask along the way, how to evaluate vendors, and our top 20 tips to make the most of your pen test based on almost two decades of experience and thousands of engagements.

In this eBook, we'll explore:

- How to establish goals and desired outcomes that align with your organization's overall security strategy
- Applying remediation techniques and implementing operationalized results
- Tips for vetting penetration test vendors that fit your needs

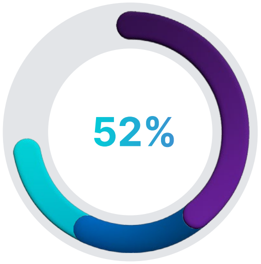
We'll include our top recommendations to make the most of application penetration engagements including pre-engagement work you can do, environmental factors to consider during testing, and shaping the post-engagement report walk-through for better secure application design in the future.



56% of the largest incidents in the last five years are tied to web application security issues, constituting **$7.6 billion in recorded financial losses.**

254 days

is the average time-to-discovery for incidents involving web application exploits — **significantly higher than the 71-day average across all event types.**



52% of events are attributed to organized criminal groups, with **23%** and **20%** respectively attributed to employees or contractors and state-affiliated threat actors.

Source: Cyentia Institute. (2022). The State of the State of Application Exploits in Security Incidents.

Getting Started

START WITH USE CASES

What are Your Goals & Desired Outcomes?

First things first: document the goals of your penetration test, including your core use cases. Here are some of the most common use cases we see in our engagements:

SECURITY COMPLIANCE

If you're seeking security compliance to bring your application to market, you can expect longer engagements, comprehensive testing, and traditional reporting at the end of your engagement.

DEVELOPMENT INTEGRATION

If you're looking for penetration testing during the development process, expect much shorter, more frequent engagements focused on integration-type tests and manual test cases. During development, look for real-time reporting; often, you'll see this as findings being delivered into bug trackers.

STRATEGIC PARTNERSHIPS, MERGERS, & ACQUISITIONS

If you are entering partnerships with external organizations or taking the plunge into an official acquisition, you need to understand the risks of inheriting new applications. Aim to get clear visibility into your expanding security ecosystem and how to steer clear of pitfalls that put your organization at risk.

THIRD-PARTY APPLICATIONS

If you have plans to incorporate third-party applications into your existing security infrastructure, penetration testing can set you up for success by exposing risks ahead of time and mapping remediation options before it is too late. You can expect to get a comprehensive view into the risks associated with integration of third-party applications.

ALIGN STRATEGY & TACTICS

How Does Penetration Testing Fit into Your Overall Application Security Strategy?

ALIGNING YOUR TOOLS AND TACTICS

Certain tools, such as static application security testing (SAST) and dynamic application security testing (DAST), are natural companions to each other. So, too, are DAST and penetration testing.

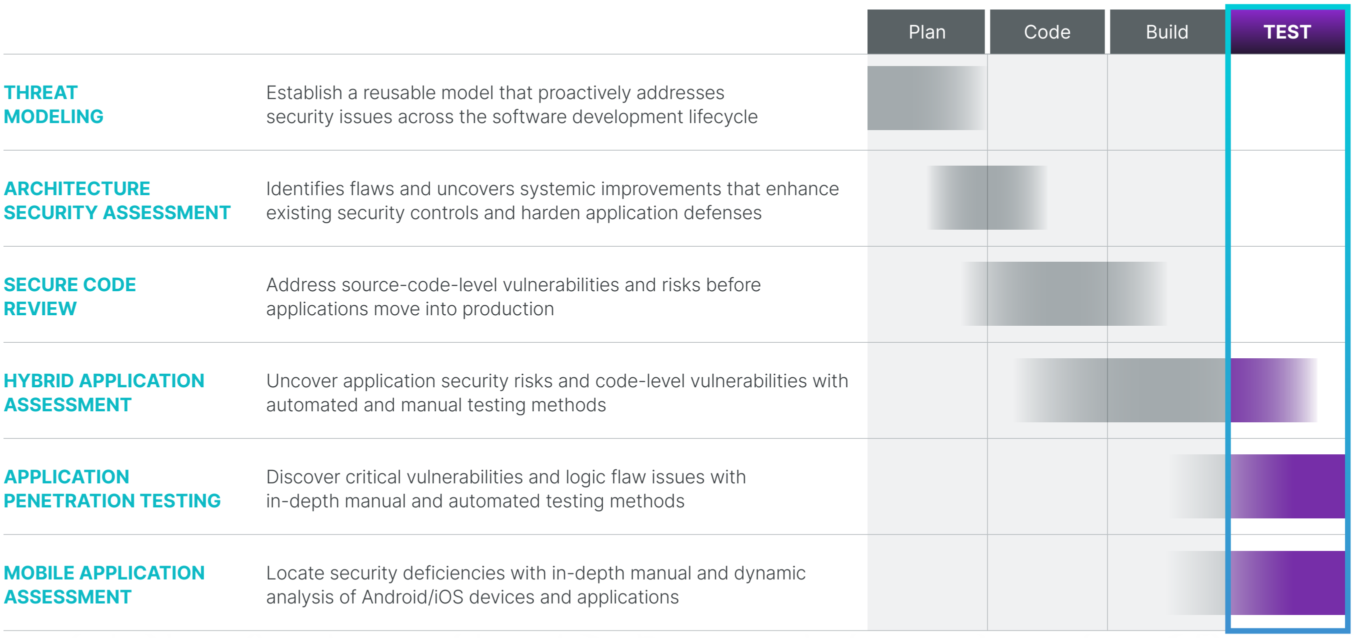
The continuous nature of DAST pairs with penetration testing because a finding from a DAST scan can be used by the penetration tester to make their job more efficient. It's a huge waste of time and penetration tester talent to be reporting the same exact vulnerabilities that your automated tools are. The key is to use penetration testing to focus on the flaws that automated tools won't find, as well as validate any findings from automated tools to determine their severity and potential business impact.

GETTING THE TIMING RIGHT

When to engage a penetration testing team depends a lot on your software development life cycle. Many will choose to proactively bring in a penetration testing firm during functional or QA testing.

Other ideal times to tap into a penetration test would be alongside your DAST dynamic scans — in fact, you can even leverage those dynamic scans as a data point for your penetration tests. Or you might choose to continually run manual tests during development as soon as a functional version of the app exists. All these strategies enable penetration testing to evolve from being a production blocker to becoming a key part of the go-to-market process.

Below is a helpful guide to better understand the full spectrum of application security from planning to testing phases and where typically penetration tests fall within this time frame.



THREAT MODELING	Establish a reusable model that proactively addresses security issues across the software development lifecycle
ARCHITECTURE SECURITY ASSESSMENT	Identifies flaws and uncovers systemic improvements that enhance existing security controls and harden application defenses
SECURE CODE REVIEW	Address source-code-level vulnerabilities and risks before applications move into production
HYBRID APPLICATION ASSESSMENT	Uncover application security risks and code-level vulnerabilities with automated and manual testing methods
APPLICATION PENETRATION TESTING	Discover critical vulnerabilities and logic flaw issues with in-depth manual and automated testing methods
MOBILE APPLICATION ASSESSMENT	Locate security deficiencies with in-depth manual and dynamic analysis of Android/iOS devices and applications

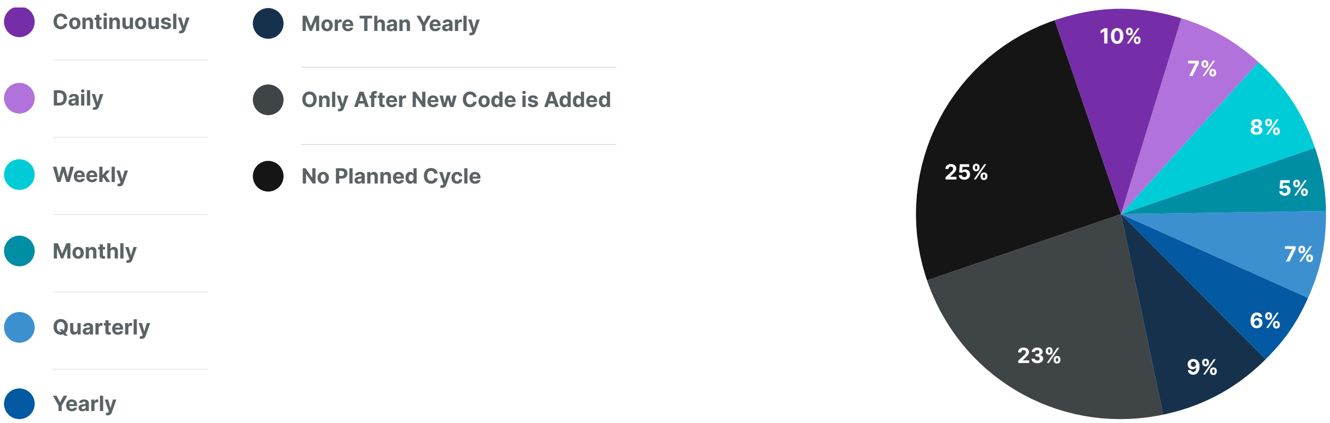
CHOOSING THE RIGHT APPROACH & FREQUENCY

Penetration testing that requires a 3-4-week lead time cannot keep up with the speed of development and production in a DevOps environment. While some of the self-scheduling penetration test solutions might sound like a good fit, they don't always scale the way you need them to.

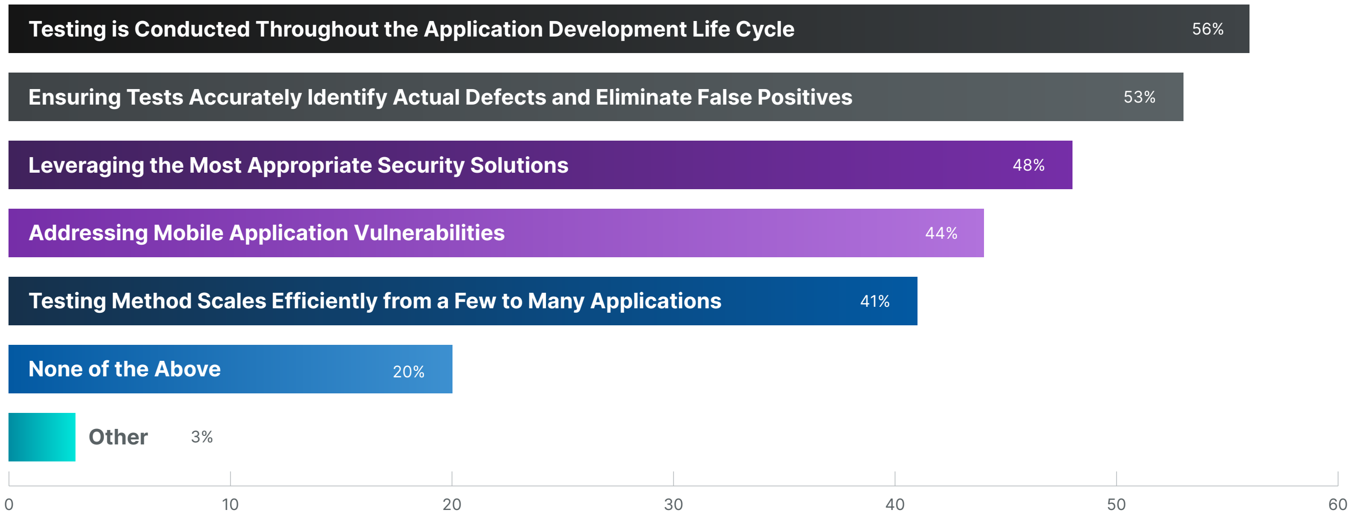
To adapt to the complexities of different schedules and testing routines within your development team, you need to make sure you have a human to engage with from the start, rather than bringing on an "automated" penetration testing service. To ensure you find the right partner, ask prospective firms if they are able and willing to customize their scheduling to fit your organizational needs or offer a flexible, more continuous service model.

Below is a quick look from Ponemon's Application Security in DevOps Environment Report that showcases how often organizations typically test their applications and what steps are taken to do so.

WHAT BEST DESCRIBES YOUR ORGANIZATION'S APPLICATION TESTING CYCLE?[1]



WHAT STEPS ARE BEING TAKEN TO TEST FOR VULNERABILITIES IN APPLICATIONS?[1]



[1] Ponemon Institute. (2020). Application Security in the DevOps Environment.

CAPTURE KEY REQUIREMENTS

How Will You Evaluate Penetration Testers to See if They Are a Good Fit for Your Environment?

CHOOSING THE RIGHT TESTING PARTNER

For the best penetration testing engagement, validate that your testers are qualified, have the experience you need, and are trustworthy. After all, you're asking them to responsibly "break" your application.

Also, consider how they will engage with your developers. The best penetration testing firms are those that are completely transparent about whom you'll be hiring (who your primary tester is) and what level of experience they have. Any avoidance to discuss these details is a red flag.

DON'T FALL FOR THE BAIT AND SWITCH

Many vendors may try to bait and switch you by bringing on a highly qualified senior penetration tester for an initial call, but then assign your engagement to a tester that's newer or, from their perspective, cheaper. When assessing security vendors ask for resumes and request interviews with the penetration testers. Remember, you're looking to hire specific penetration testers that fit your needs, not just assessing the testing firm overall.

USE YOUR EXISTING COMMUNICATION CHANNELS

How do you want your testers to communicate with your team during the engagement? If you typically use a Slack channel or Microsoft Teams to communicate internally, ask your testers to communicate through those same channels, provided they can do so securely. The goal is for security testers to adapt to the developer's needs, not the other way around.

PRO-TIP

Consider giving the penetration testers access to the same bug tracking systems your development team uses. This way, the team will start to see the penetration test as another testing tool in your pipeline instead of a separate activity done later in the process. These subtle changes help shift the mindset towards a DevSecOps approach. With this approach, penetration testing can adapt to business needs where it counts: experience, speed, scale, and continuous feedback.

INSIST ON MEANINGFUL RECOMMENDATIONS AND RETESTING

A penetration test is not complete without a remediation plan. You could have the best penetration tester in the world, but if they give your developers a list of high and critical findings without a pragmatic set of recommendations to remediate, you're not setting them up for success. Make sure your remediation process also includes a retest to validate and verify "fixes" have successfully addressed root cause issues. Equally important, ask your security tester for recommendations to get to the bottom of the issue with the goal of repetitive secure application design in the future. Penetration testing is just one component to securing applications – threat modeling and architecture security assessments (ASA), for example, ensure applications are built to be less vulnerable and meaningful recommendations from testers can go a long way towards achieving this.

ENSURE AGILITY

How Will Your Penetration Testing Adapt to Modern Environments?

CONSIDER THE TESTING ENVIRONMENT

Penetration testing is vastly different depending on where the development environment resides. Consider where the testing will take place – the cloud, on-premises, or both. The penetration tester will need to access your organization's systems, like containers for example, so planning for intricacies can ensure a smoother engagement.

GOING BEYOND AUTOMATION

To work in modern environments, penetration testing should be focused on opportunities to creatively find footholds and flaws in your applications that cannot be found by automated tools. Tools and automation are good at identifying common, known issues, but they can't think, and they will not seek out new pathways for an attack beyond surface-level findings.

WHERE TO START

Start with creating integration tests that are short, fast, and repeatable. Model how your QA teams currently do functional testing and implement some of those aspects into your penetration test. For example, you could integrate specific tests designed to uncover business logic flaws or authentication vulnerabilities. As a free resource, The OWASP Web Security Testing Guide Checklist is a great place to start developing manual test cases.

GO BEYOND AUTOMATION. WHICH APPLICATION FLAWS ARE MISSED BY AUTOMATED TOOLS?

Make sure your application security testing plan captures as many vulnerabilities as possible. Here are a few common vulnerabilities that many tools miss but can lead to massive risks if they're undetected before deployment:

- Business logic
- Privilege escalation
- Insecure Direct Object Reference (IDOR)
- Subdomain takeovers
- Blind XXE injection
- Complex DOM based XSS
- Session management
- Missing authentication for critical functions
- Missing authorization
- Reliance on untrusted inputs in a security decision
- Execution with unnecessary privileges
- Incorrect authorization
- Improper restriction of excessive authentication attempts
- Use of a one-way hash without salt
- Password reset and user management functionality
- File upload flaws

APPLY REMEDIATION RECOMMENDATIONS

How Will You Operationalize Results?

REVIEW AND DISCUSS THE TEST RESULTS

Penetration testers often provide thorough reports with invaluable information that consists of several elements. An executive summary will list a rundown of the steps that were done during the test. From there, different tests offer varying information, but generally, there are details about the findings the penetration testers gathered during the process. Results also usually include a master list of issues that need to be addressed, and at least a basic list of recommendations. Testers are often willing to answer questions, even after the test has been completed, and provide further insights and recommendations.

Since penetration testing reports show how testers exploited your infrastructure, organizations can consider not only the initial findings, but they can also do further analysis to get to the root cause. Finding out the what the real risks are is a key part of remediation.

DEVELOP A REMEDIATION PLAN AND VALIDATE IMPLEMENTATION WITH A RETEST

A single penetration test should serve as a baseline. An integral part of penetration testing strategies is to retest frequently against that baseline to ensure improvements are made and security holes are closed. Penetration test results often come with a hefty to-do list, which means it's unlikely that every single weakness can be fully addressed right away. A penetration test postmortem should carefully consider how to prioritize what needs to be addressed.

Before scheduling the next penetration test, it is helpful to review exactly what penetration tests were run previously. By considering whether additional or different tests should be completed, you can ensure you're getting the most valuable insights possible.

INCORPORATE FINDINGS INTO YOUR LONG-TERM SECURITY STRATEGY

Continue to put your organization to the test on a regular basis. Penetration testing should be conducted frequently to ensure you're continuously reducing your cyber risk exposure. The goal of penetration testing shouldn't be to earn a passing grade, but rather to utilize the results to elevate your overall cybersecurity posture and the maturity of your security program to ensure your organization is as impenetrable as possible.

PRO-TIP

Fast Feedback for Penetration Testing

Fast feedback is important when it comes to application penetration testing; flaws identified through a penetration test should be given to application developers immediately. Penetration test findings can always be accompanied by screenshots, walk-throughs, and even video demonstrations of how a particular flaw was exploited. A walk-through provides an opportunity for penetration testers to prove that the findings are real and could be exploited in a real-world environment. Traditional reporting won't cut it in an agile DevOps environment. Prove it, or it doesn't exist.

On The Hunt: Evaluating Penetration Test Vendors

As you research penetration testing vendors, seek out firms who have a clear understanding and respect for DevOps, and even better – DevSecOps. Select those that prioritize modern penetration testing over the legacy testing models.

We also recommend finding a testing partner that understands how to balance manual and automated testing approaches. Automated solutions can be used to uncover low hanging fruit, while manual testing can both confirm those results and uncover flaws that automated tools cannot. Notably, these types of flaws (e.g., IDOR, business logic, etc.) are the source of countless data breaches. The key is to find the right balance between automation and validated testing to achieve the right outcomes - reliance on one or the other can lead to sacrificing scale or thoroughness.

And push for a firm that allows you flexibility in how they report their findings. Though many providers still only provide PDF reports of findings, more progressive firms offer customization, as well as findings that can be pushed directly into vulnerability management systems.

CRITICAL CRITERIA FOR CONSIDERATION

- Core business competency
- Years of experience
- Quantity of engagements
- Breadth of portfolio
- Innovation & research
- Accreditation
- Demonstration of delivery
- Client testimonials
- Industry publication & recognition
- Adherence to regulatory standards & industry frameworks like OWASP
- Tester experience, certifications, & tenure
- Training & development

As you evaluate potential penetration testing vendors, ask pointed questions to better understand if they are the right fit. We recommend:

- What certifications are held by your testers?
- What is your penetration testing methodology and what standards/methodologies do you follow?
- What is covered in your penetration testing report?
- How do you maintain internal security in your company?
- Does your penetration testing service include remediation recommendations or detailed guidance?
- Have you made any vulnerability disclosures recently?
- Is your penetration testing service automated, manual, or a mix?
- Who would be conducting our penetration test, and what are their qualifications?
- Do you perform background and screening checks of your team members?
- Will my services remain available during a penetration test?
- What vulnerabilities and weaknesses are covered?
- Is there an option to customize to test a particular scenario?

PRO-TIP

Seek penetration testing partners that combine automation with continuous, expert-driven penetration tests. This hybrid approach allows the security provider to scale their testing capabilities without losing the human ingenuity that enables them to think like an attacker.

20 Tips to Make the Most of Your Testing

Spending money on penetration tests is an investment in your application, so you want to ensure you're getting your money's worth. However, there are several common pitfalls that can cost you in terms of quality, project delays, or unnecessary expense. Here is a list of our top tips to ensure your engagement is successful.

01 CONDUCT YOUR OWN PRE-ASSESSMENT

If you have the staff, consider performing your own in-house assessment prior to contracting a penetration test or opening a bug bounty program. This will help eliminate the low-hanging fruit (i.e., bugs that are easily detected with automation and scanning).

This can be especially important with bug bounty programs because having to pay out for many easy-to-find bugs could cost more money in bounties than the allocation of a fulltime resource.

Eradicating these vulnerabilities in advance will allow you to rely on the professionals for the harder to find bugs.

02 KNOW YOUR ASSESSMENT GOALS

Determine specific goals and trophy targets. As with any project, clearly stating goals for the assessment ahead of time helps keep everyone on track and allows the team to prioritize vulnerabilities surrounding your greatest concerns.

Put Yourself in Their Shoes

If you are a product company, unauthorized access to schematics/design documents, unreleased marketing material, or any other information that could be at risk for corporate espionage might be the primary goal for the assessment team. While the team will still test other functionality and produce any other findings encountered, a primary goal will provide a clear focus for the penetration test.

03 AIM FOR ACCURATE SCOPING SURVEYS

It is important to describe the size and scope of the application as accurately as possible. Scoping teams will often provide a survey for your team to fill out to describe various aspects of the target(s) being assessed. It may take a little more time upfront, but it will also ensure that the project's assigned hours are accurate as well, thereby setting the project up for success from the start.

Overestimating (or over-scoping) poses fewer risks because your penetration test team can always dig deeper into any application or reallocate the hours for a different testing activity. Reporting an accurate (or even slightly overestimated scope) is the first step to ensuring project success.

OVER-SCOPING

When determining the line of code (LoC) count for a source code review, be sure to remove test cases from repositories before running automated tooling. Let's say you're running a tool like cloc **(https://github.com/AlDanial/cloc)** to provide a source code estimate. Although it will automatically subtract comments and blank lines, which is helpful, it cannot distinguish between test cases and product source code cases. This could mean that a result of 400k LoC could include 100k LoC of test cases, overestimating the source code count by 25%.

UNDER-SCOPING

If you have a web application and you guess that it had 50 endpoints when it in fact has 100, then the test will be under-scoped. Under-scoping may require a last-minute change order which could mean more hours, causing budget issues, project delays, or interference with other deadlines. Going ahead without a change order means you will end up with a more limited test than you planned.

04 CONSIDER A MULTI-TIERED ASSESSMENT

While it's common to test the outward-facing portion of your application, a multi-tiered assessment can help ensure strong detection and defense mechanisms after various levels of compromise and will lead to more robust application security. This form of assessment provides the testing team with authenticated access to various levels of the underlying architecture directly, as opposed to requiring a code execution to be discovered.

WEB APPLICATION ASSESSMENTS

In a multi-tiered web application assessment, the team might assess the application as various user roles (as you would find in a standard assessment), but then they would also simulate the compromise of customer service users, an application server, or a back-end server.

These roles could be set up as the following:

- No access/public sign-in
- Application user roles (e.g., users, organization users, and administrators)
- Customer support user
- Command-line access to a primary web service
- Command-line access to a secondary back-end service

Attempting attacks from these privileged spaces allows the network monitoring team to become familiar with what malicious behavior looks like, and it allows security at the server level to be evaluated. For most organizations, security at this level is not formally evaluated until a break-in occurs or until privileged access is obtained during a penetration test.

05 DISABLE YOUR WAF DURING TESTING

Why should you disable your web application firewall (WAF) during a security test after spending all that money on it? It's the same reason a patient helps a doctor by pulling up their sleeve when looking for chicken pox. Disabling the WAF is the fastest and most time-effective way to diagnose issues in the underlying application. If you disable the WAF, it allows the team to focus on identifying flaws in your application, instead of flaws in third-party appliances. Don't give the WAF a free penetration test.

That said, if you're concerned about the efficacy of a WAF in relation to your application, coordinate with the team to re-enable the WAF toward the end of the assessment. That way, they can look for specific bypasses to determine when the WAF may be effective in stopping an attack.

Alternatively, consider having two test environments: one with defense-in-depth controls (e.g., a WAF) and one without. This allows the team to discover application vulnerabilities without spending excess time bypassing filters. In our experience, WAFs slow down attacks more than they prevent them.

06 DISABLE RISK-BASED SESSION EXPIRATION

Disable application features that may interfere with testing, such as session expiration associated with malicious payloads. While this feature may slow down attackers in production, it will also slow down your penetration testers and limit the number of tests performed per billable hour.

07 ENSURE A STABLE, RESPONSIVE TEST ENVIRONMENT

For the most effective penetration test, ensure that the test environment is just as responsive, complete, and stable as the production environment. To ensure stability, don't alter the test environment during a penetration test. If the environment is altered, it can result in missed findings due to downtime or false positives from in-progress bug fixes.

Downfalls of an Unstable Testing Environment

In worst-case scenarios at Bishop Fox, we've seen test environments with five to ten seconds of latency request. We've also had penetration tests conducted through screen control on WebEx. Believe us, there are better solutions for a testing environment.

08 FILL THE TEST ENVIRONMENT WITH DATA

Do not provide an empty test environment. Fill it with test data to allow consultants to demonstrate authorization bypasses, such as gaining access to another user's files. Without this data, it will be more challenging to validate findings, and the final report may lack strong examples of business impact.

Some customers mirror production data to the test environment, while others fill it with QA data. If you like, you can add specific trophy files or data to the environment for us to focus on obtaining.

09 ENSURE DEV TEAM AVAILABILITY DURING THE TEST

Lack of product team availability is a commonly overlooked risk to successful projects. Penetration tests often involve discussions with development or security team members to strategize solutions. When these team members aren't available, project delays can occur, and the assessment team can be limited in providing tailored remediation recommendations.

As a result, before scheduling a penetration test, confirm that your development, security, and any other essential team members have availability on their calendars and are not out of office.

10 CONFIRM ON-TIME PRE-ENGAGEMENTS

Most testing firms will provide a list of access requirements in advance that are necessary to test your application. To avoid wasting time, deliver pre-engagement requirements on time. Feel free to reach out to your consulting team and ask them to confirm access before testing is scheduled to begin to ensure an on-time start.

For instance, access requirements might entail provisioning multiple user roles for an application. However, there are multiple ways access might be incomplete. Perhaps the accounts were provisioned, but they were linked to an employee email account instead of a tester's email account. Or maybe the accounts were added, but the team can't test the application because the test environment gateway needs to whitelist the team's IP addresses. All of these issues can lead to delays or slow down a test, taking away valuable testing hours from a project. Bottom line: it's always good to confirm access prior to the start of testing.

11 IF POSSIBLE, PROVIDE SOURCE CODE

Source code is always better than no source code. Even if you are purchasing a black-box penetration test, providing source code allows the team to track down issues faster and identify more vulnerabilities. No penetration tester will reject source code.

12 PROVIDE TEST SUITES & DEV TOOLS

The more information you can share, the better. In addition to source code, provide any QA/dev tools (e.g., Postman collections, custom dev tools, and test data) that might allow the assessment team to more effectively interact with, compile, or test your application. This will also reduce the amount of time the consultants need to construct preliminary test cases.

13 PROVIDE DEV & CUSTOMER DOCUMENTATION

Provide any developer and customer-facing documentation or diagrams. Like onboarding a new developer, it reduces the time the consulting team requires to gain a baseline understanding of the application's architecture.

14 TEST THE SECURITY, NOT THE OBSCURITY

If your application relies on any obfuscation or anti-debugging, disable that obfuscation during the test, unless you want to focus on assessing the efficacy of the obfuscation instead.

These tactics are typically used to slow down an attacker, which, while valuable in an attack scenario, may incur an unnecessary cost when assessing your application's security.

15 ASSIGN A RESOURCE TO RESOLVE BLOCKERS

Remember, like lawyers, consulting firms track billable hours. Make the best use of the time you're paying for by assigning a resource from your team to resolve any blockers that might emerge, which will allow the assessment team to solve problems faster.

While some clients choose to provide consulting teams with an email distribution list to resolve issues, an assigned project manager can ensure a quick turnaround and use internal escalation paths to expedite resolutions, which is much more effective.

For example, if the assessment team is missing credentials, a delayed response could significantly affect the team's ability to test. For this reason, consider assigning a specific resource to unblock your consulting team and ensure information or access requests are fulfilled in a timely manner.

16 MAINTAIN OPEN COMMUNICATION WITH TESTERS

Consider creating a Slack channel (or any other instant messaging platform) with your development team where the assessment team can ask questions or request information. You might also have the resource assigned to blockers that serve as a conduit to ensure your team is responding within a few hours.

17 ESTABLISH AN ESCALATION PLAN FOR HIGH-RISK FINDINGS

Have a plan in place for handling critical- and high-risk vulnerabilities. Ensure the relevant development teams are aware of this possibility so they can be prepared to triage any high-risk issues as they are reported. Keeping everyone in the loop and having your various teams prepared to push a new release will eliminate unforeseen chaos.

18 SCHEDULE TESTS DURING THE SUMMER

Many security teams find themselves rushing to spend unused budget dollars at the end of the year. As a result, consulting firms are busiest in November and December, so you might not get your penetration test on the calendar. Instead, test in the summer, which will afford you more testing flexibility, more diverse availability of resources, and more opportunities to extend timelines or schedule follow-up assessments.

19 USE PENETRATION TESTERS FOR PENETRATION TESTING

Consultants that specialize in penetration testing may be great at finding vulnerabilities in your application, but may not be skilled at delivering training on how to do secure development.

To ensure a successful assessment, scope the project correctly from the start so that you have the right resources to successfully complete each of the items described in the statement of work. Education may be something a consulting firm can offer, but it will need to be considered during the initial scope and may require different consultants; this also reduces project risk (e.g., causing delays due to staffing changes or additional costs).

20 ASK QUESTIONS DURING THE REPORT WALK-THROUGH

Here are Some Questions that May be Helpful to Ask:

1. After we remediate these findings, how will you feel about the security of this application?
2. Did you feel like you got a thorough view of the application? If not, what would you have wanted to test further?
3. What should we focus on for our next penetration test? What functionality or feature concerns you the most?
4. Are there any strategic design changes that you would recommend?
5. How can we have our QA team test for issues like ____________________ to avoid them in the future?
6. Are there any automated tools that we should consider adding to our CI/CD pipeline?
7. We are considering migrating to the ____________________ service/platform/framework. What things should we consider during this migration?
8. I noticed there weren't many (or any) findings on the ____________________ feature. What were your observations during testing?
9. Did you have any blockers or delays during testing? If so, what can we do to reduce those in the future?
10. How can I stay up to date on security risks for ____________________? Are there any projects, newsletters, or news sources that my team should consider monitoring?

About Bishop Fox

Bishop Fox is recognized as the leading authority in offensive security, providing solutions ranging from continuous penetration testing, red teaming, and attack surface management to product, cloud, and application security assessments.

Over the past 16 years, we've worked with more than 25% of the Fortune 100, 8 of the top 10 global tech companies, and hundreds of other organizations to improve their security. Our award-winning Cosmos platform was named **Best Emerging Technology** in the 2021 SC Media Awards and our offerings are consistently ranked as "world class" in customer experience surveys.

Security isn't just a job to us. We do this because we love it — and because we're committed to the common good. In fact, we have authored 15 open-source tools, shared groundbreaking research, and published more than 50 security advisories in the last 5 years.

Application Security Services

Not all application testing is created equal. From application design processes to deep source code review, we'll help you build safer apps, ensure compliance, and level-up your DevSecOps.

Application Penetration Testing

Our award-winning, in-depth application penetration testing goes well beyond discovering vulnerabilities to analyze the inner workings of your applications and identify critical issues, exposure points, and business logic flaws.

Hybrid Application Assessment

Dissect every aspect of your app's security with source-code-assisted application penetration testing that uncovers a broad range of vulnerabilities and exposures.

Mobile Application Assessment

Put your mobile apps to the test with in-depth static and dynamic analysis across iOS and Android devices that proactively identifies attack vectors and risks, including weaknesses across code, services, APIs, and more.

Architecture Security Assessment

Put your applications and underlying security architecture under the microscope to illuminate critical flaws and identify systemic improvements that will enhance security controls and harden defenses.

Secure Code Review

Improve the overall security of code and eliminate flaws that fall into production using a combination of automated review and detailed human inspection that uncovers the full spectrum of security flaws, vulnerabilities, and business logic errors.

Threat Modeling

Proactively address security issues across the software development lifecycle with in-depth analysis of application design, threats, and countermeasures that become integral to ongoing DevOps processes.

CONNECT WITH US

Get started today.

Are you ready to start "defending forward"? Get in touch with our offensive security experts today to explore solutions that meet your unique business needs.

Request a Meeting   Explore Cosmos

8240 S. Kyrene Rd. • Tempe, AZ 85284
480.621.8967
hello@bishopfox.com • bishopfox.com