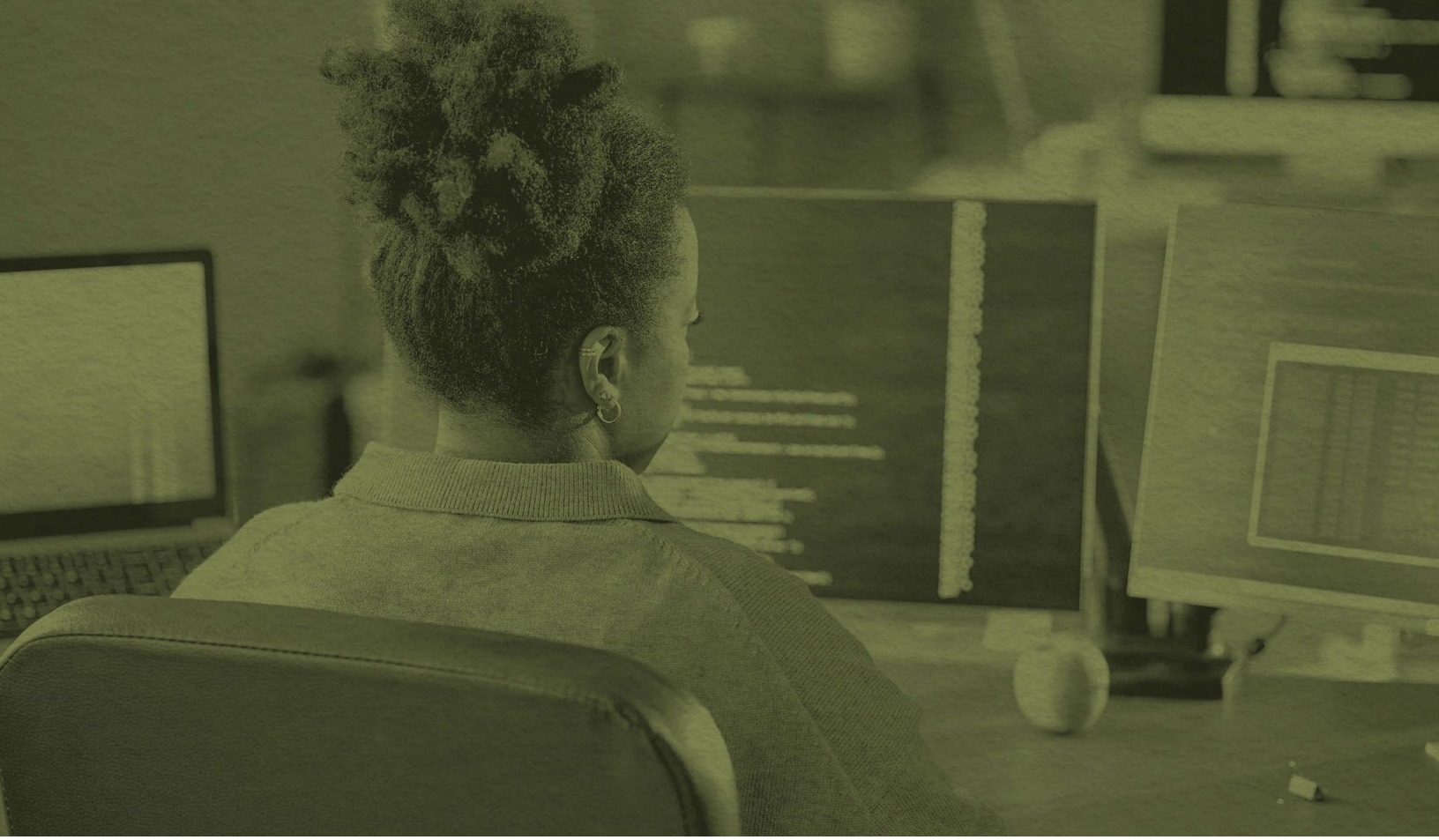

NetSPI
PROACTIVE
PROTECTORS
THE APP
INTRUDERS
How to Build a Winning
Application Pentesting Program

NetSPI™

Table of Contents



Introduction

How many apps do you interact with on a daily basis? Think about it. Today's world runs on applications. Data from **RealityMine** show that on average, US smartphone users interact with 18 different apps every day, making them a prime target for cyberattacks.

Types of applications span from web and mobile to thick and virtual, to name a few. All of these applications are developed with distinct code and unique functionality. Add this individualization to the velocity of using AI to vibe code, and the reality becomes an incredibly diverse landscape of applications — all of which need to be secure.

Creating an application pentesting program is a feat. Fortunately, the team at NetSPI has walked this path for 20+ years and laid the foundation for a strategic approach to securing your applications.

In this comprehensive guide, we'll learn from NetSPI's Application Pentesting service line directors about how to approach application pentesting with a sound strategy that reduces risks, enables innovation, and ultimately protects your business. The considerations our team shares meet the intricacies of an expanding application portfolio to ensure your business stays resilient in the face of today's constant threats.

Why We Need a Better Approach to Application Pentesting

Trying to address every single risk in your application ecosystem to achieve 100% security is a surefire way to fail. This ideal end-state sounds appealing to the untrained ear, but ask anyone in cybersecurity about the state of remediating vulnerabilities across their applications, and you're sure to get a heavy sigh.

The sheer volume of vulnerabilities today requires a tailored prioritization strategy, often beyond Common Vulnerability Scoring System (CVSS) scores, and into asset dependencies in light of business context. For example, securing a critical flaw in an application that isn't connected to any internal systems is likely less valuable than securing a medium-level risk in an application that stores sensitive data.

Pentesting programs have traditionally relied on periodic testing to meet compliance mandates. However, this testing cadence only gives a glimpse into the state of security at a specific point in time. The volume, breadth, and complexity of applications today require an ongoing testing approach that combines automated tools with manual deep dives to effectively protect your systems.

Comparison: Ongoing versus Point-in-Time Pentesting

Aspect	Ongoing Testing	Point-in-Time Testing
Operational Approach	Integrates automated testing directly into the development lifecycle (CI/CD) for continuous feedback.	Utilizes periodic, in-depth manual penetration testing focused specifically on critical applications.
Strategic Alignment	Validates security investments in real-time, building true organizational resilience as you scale.	Provides necessary assurance for high-risk assets to satisfy specific compliance mandates and audit requirements.
Risk Management	Proactively identifies vulnerabilities during development, preventing security debt before it reaches production.	Uncovers complex business-logic flaws and critical exposures that automated scanning often misses.

Because the world runs on applications, it's safe to say that a business' livelihood relies on its application security. The stakes of a data breach are high: lost revenue streams, diminished customer trust, negative brand perception. Meeting security regulations because they're required is one thing. But creating digital resilience through a modern application pentesting program is what delivers actual peace of mind.

$4.4M

The global average cost of a data breach

IBM's Cost of a Data Breach Report 2025

Not All Risks Are Created Equal

When evaluating the frequency of your application pentesting, consider the risk classification of your applications. For example, an external-facing app that stores sensitive data has a higher level of risk than an internal application with educational content. Evaluating these factors for each of your applications will help you determine the ideal frequency and depth of testing.

In 2025, NetSPI found that more than 90% of critical risks with a CVSS of 9 or 10 are missed by scans, making it advantageous to use a manual approach to verify vulnerabilities.

Quick Guide: Recommended Application Pentesting Frequency

Risk Level	Testing Frequency	Examples	Additional Testing Triggers
Critical / High-Risk	Quarterly to Semi-annually (every 3-6 months)	• Customer-facing banking/payment apps • Healthcare systems with protected health information (PHI) • Systems processing credit card data • Administrative/privileged access portals	• Major releases or architecture changes • After security incidents • New regulatory requirements • Significant third-party integrations
Medium-Risk	Annually (every 12 months)	• Internal business applications • Customer portals with limited personally identifiable information (PII) • B2B partner platforms • E-commerce sites • Customer relationship management (CRM) systems	• Significant feature additions • Technology stack upgrades • Change in data classification • Post-incident validation
Low-Risk	Every 2-3 years or on major changes	• Internal knowledge bases • Read-only dashboards • Marketing websites • Internal tools with no sensitive data • Archived/legacy systems	• Platform migrations • Change from internal to external access • Addition of authentication/data storage • Compliance requirement changes

Note: These are baseline frequencies. Regulatory requirements, such as PCI-DSS, HIPAA, SOC 2, may mandate more frequent testing.

The key to a successful application pentesting program is a trusted partner that can adapt the scope of testing based on your unique application. This requires the right tools and deep domain expertise to create the ideal mix of automated and manual analysis. Automated pentesting tools are extremely useful — even the AI-only ones. But AI alone is not enough.

A modern application pentesting program relies on humans to validate what the tools are telling them, and to use creative methods to approach attacking an application, just as a hacker would.

Securing LLMs and AI-Integrated Applications

We can't talk about modern pentesting without talking about AI. Love it or hate it, AI has exploded the security considerations for digital environments. This new landscape has introduced novel risks and vulnerabilities in applications. **Common ones our security experts see include:**

Prompt Injection and Model Manipulation

Crafting effective prompts that can influence the behavior of the model, resulting in outputs that were not intended by the model.

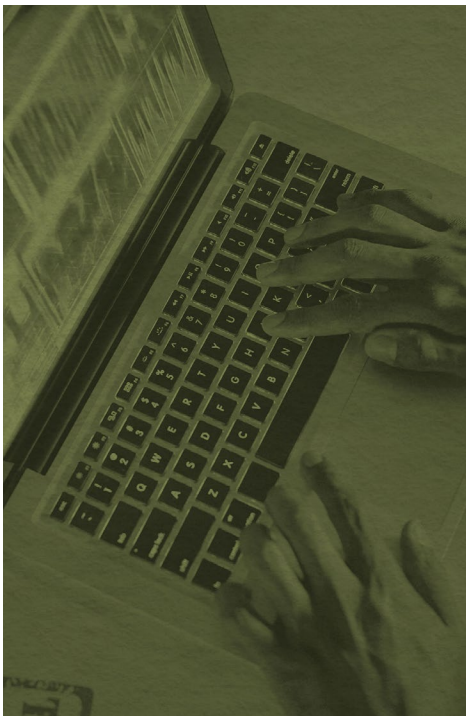
Data Poisoning and Training Data Compromise

The deliberate manipulation of the training process to compromise the performance or behavior of the target machine learning.

Interference/Inversion (Exfiltration)

The act of interfering/reflecting sensitive information about or from the training data.

A key theme throughout building your application pentesting program is having the right expertise for the test. AI has evolved the skillset pentesters need today, and tenured security testers, such as NetSPI's, stay on top of the latest research — and even help develop the standards for AI testing. See **How Microsoft and NetSPI Partnered to Build a Standardized AI Security Framework Securing 70+ Products**. This level of expertise helps you rest assured that your AI-integrated applications stay secure as you innovate.

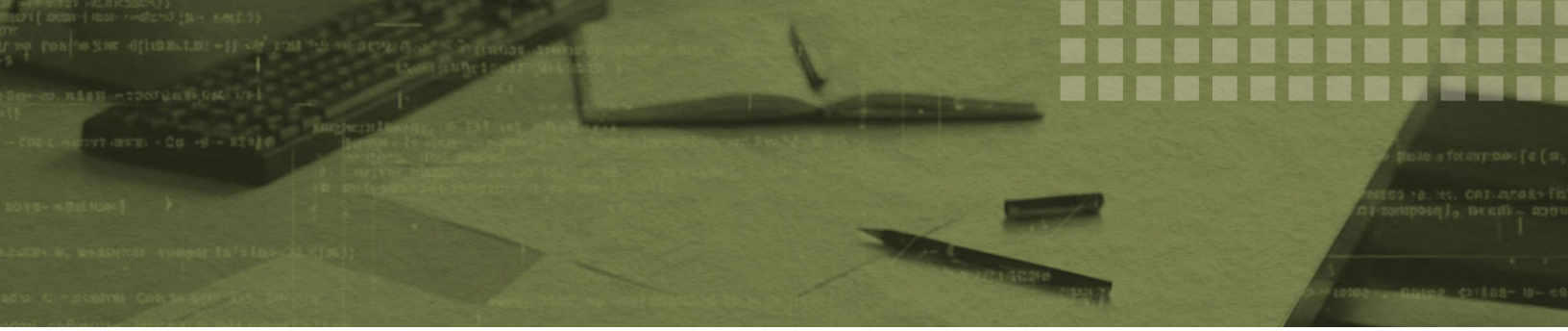


Specialized Testing Considerations

Here are a few points to keep in mind when approaching pentesting for LLMs and AI-integrated applications:

1. Evaluate the security posture of both the **application and the underlying AI model.**
2. Develop test cases that account for the **non-deterministic nature** of AI.
3. Combine traditional application security testing with **model-specific assessments.**
4. Ensure results align with the **model's intended business use case**, as well as the coinciding industry context.

The intricacies in the above considerations are what makes domain expertise so important when it comes to pentesting. Next, we'll explore considerations specific to distinct types of applications so you know what to look for as you approach your upcoming tests.



Application Pentesting Considerations

API Penetration Testing

Application Programming Interfaces (APIs) are the backbone of modern applications, enabling communication and data exchange between various services, systems, and components. They are critical for connecting applications, especially in AI-integrated environments.

Core Objective
As the connective tissue of modern digital ecosystems, APIs can expose critical business logic and data access points to potentially untrusted clients. Pentesting identifies and remediates vulnerabilities in API implementations before attackers can exploit them to compromise sensitive data, manipulate functionalities, or bypass authentication and authorization controls.

Focus Areas

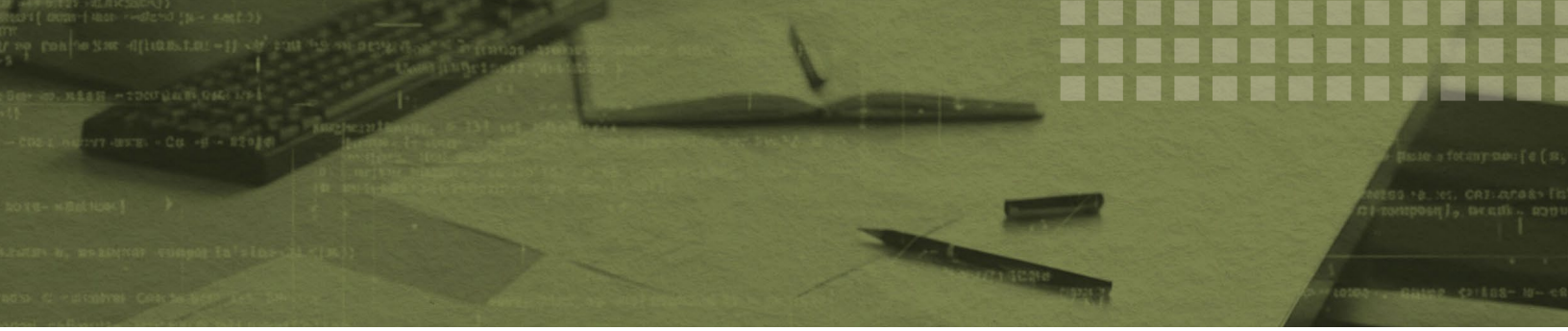
- Encryption
- Authentication & Authorization
- Input Validation
- Rate Limiting
- Business Logic

Recommended Frequency

- **Quarterly** for public-facing APIs
- **Semi-annually** for internal/partner APIs

Common Attack Vectors

- Broken access controls (both authentication and authorization)
- Third-party attacks (insecure integrations, supply chain vulnerabilities, API data consumption)
- Resource consumption (lack of throttling, excessive operational costs)
- Server Misconfigurations (TLS implementation, missing server patches, information disclosures)
- Insufficient API Inventory (deprecated APIs, cross-boundary API access)



Application Pentesting Considerations

Mobile Applications

Mobile applications are Android or iOS applications installed on a mobile device such as a smartphone or tablet. They typically interact with a server-side component, which is referred to as the API or server.

Core Objective
Mobile applications handle sensitive user data and operate with elevated system permissions, making them critical attack surfaces. Pentesting identifies and remediates vulnerabilities in iOS and Android applications before attackers can exploit them to compromise user privacy, data integrity, or device security.

Focus Areas

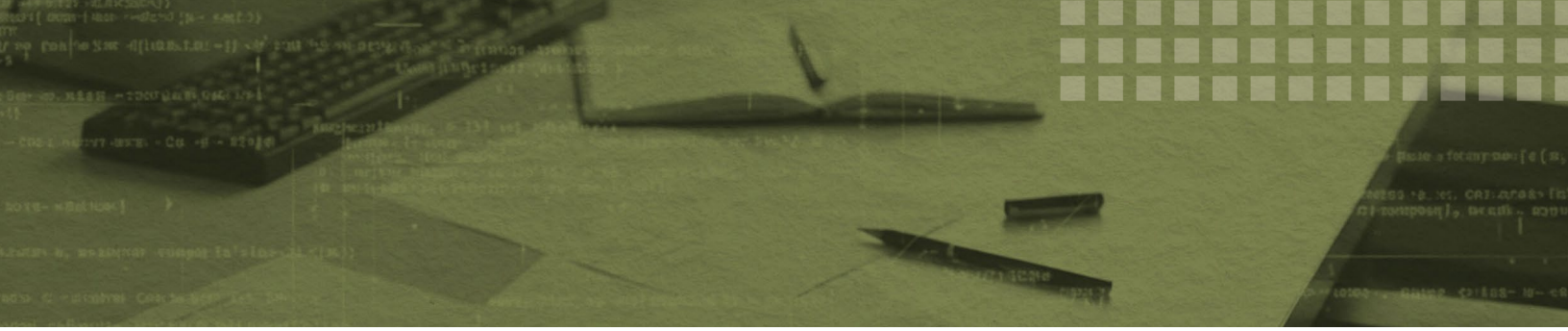
- **Dynamic Analysis**
  Authentication, Authorization, Input Validation, Privilege Escalation, Local Auth Bypass, Missing Local Auth, App Timeout
- **Static Analysis**
  Hard Coded API Keys, Hard Coded Passwords

Recommended Frequency

- **Quarterly** for apps with frequent updates
- **Semi-annually** for others

Common Attack Vectors

- Direct attack on the server (Injections, Authentication bypass, Authorization flaws
- Hard coded API keys
- User attacks
- Lost phone (Data stored insecurely, insecure credential storage, missing local authentication, local authentication bypass)
- MiTM threats (flawed TLS implementation)
- Mobile malware threats (IPC issues – exposed Content Provider)



Application Pentesting Considerations

Web Applications

Modern web applications are sophisticated software systems that combine client-side code (running in the browser) with server-side logic (running on remote servers). This distributed architecture enables the interactive experiences we rely on today, from real-time collaboration tools to streaming platforms and enterprise software.

Core Objective
Web applications must be publicly accessible to serve users, making them inherently exposed attack vectors. Pentesting identifies and remediates vulnerabilities in web applications before attackers can exploit them.

Focus Areas

- Authentication & Access Controls
- Input Validation & Application Logic
- Data Exposure & Secure Communications
- Application Architecture
- API Security

Recommended Frequency

- **Quarterly** for high-traffic or critical applications
- **Semi-annually** for others
- **Annually** for internal-only applications or those that have never been tested

Common Attack Vectors

- Broken access controls (both authentication and authorization)
- Injection attacks (SQLi, XSS, XML, SSRF, etc.)
- Business logic flaws
- Vulnerable technology (missing server patches, server misconfigurations, vulnerable libraries)
- Sensitive data disclosure (verbose errors, API key exposure, etc.)

War Story: Critical Security Flaw in Oxidized Web Application

NetSPI Red Team
Principal Security Consultant
& Cybersecurity Consultant

Discovery and Impact

NetSPI Red Team came across a critical vulnerability in the Oxidized Web application, a tool used for managing router and switch configurations. The vulnerability, tracked as CVE-2025-27590, allowed an attacker with access to the /migration page to overwrite any local file writable by the oxidized user. This could lead to remote code execution on the server.

The vulnerability stemmed from a deprecated data validation issue in the /migration page, which enabled attackers to overwrite critical files like ~/.bashrc. This could result in unauthorized access, further exploitation, and potential compromise of the server's infrastructure.

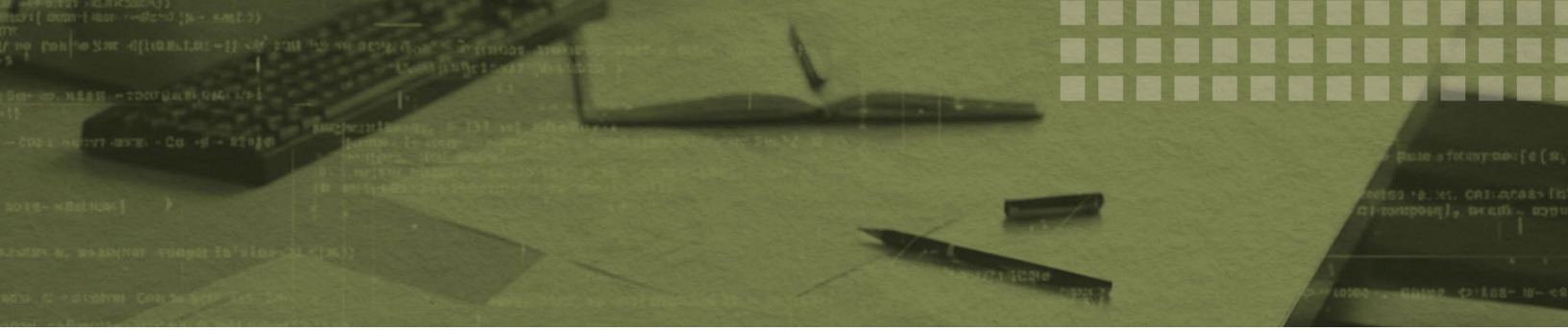
How We Did It

1. The team identified the /migration page as a potential target due to its complex data ingestion and merging functionality.
2. By analyzing the source code and testing a local instance, the team discovered that the page allowed arbitrary file writes without proper input validation.
3. A proof-of-concept attack was executed, demonstrating the ability to overwrite the ~/.bashrc file. This modification enabled the addition of an SSH key to the ~/.ssh/authorized_keys file, granting unauthorized access to the server.
4. The team confirmed the exploit by successfully logging into the server using the injected SSH key.

Remediation

We reported the vulnerability to the Oxidized Web development team, who promptly removed the deprecated /migration page in version 0.15 of the application. NetSPI recommends that organizations using Oxidized Web:

- Restrict access to the web interface to authorized personnel only
- Upgrade to the latest version (0.15 or higher) to mitigate this vulnerability



Application Pentesting Considerations

Thick Client Applications

The term “thick client” is a term generally used to describe a “desktop” application that runs on a general-purpose operating system (e.g. Windows, Linux, MacOS). Zoom, Microsoft Office, Adobe Photoshop, and the Chrome web browser are examples of “thick clients.”

Core Objective

Thick applications operate with direct access to system resources and often handle critical business functions and sensitive data locally. Pentesting identifies and remediates vulnerabilities in thick client architectures before attackers can exploit them to compromise data integrity, escalate privileges, reverse engineer proprietary logic, or gain unauthorized access to backend systems.

Focus Areas

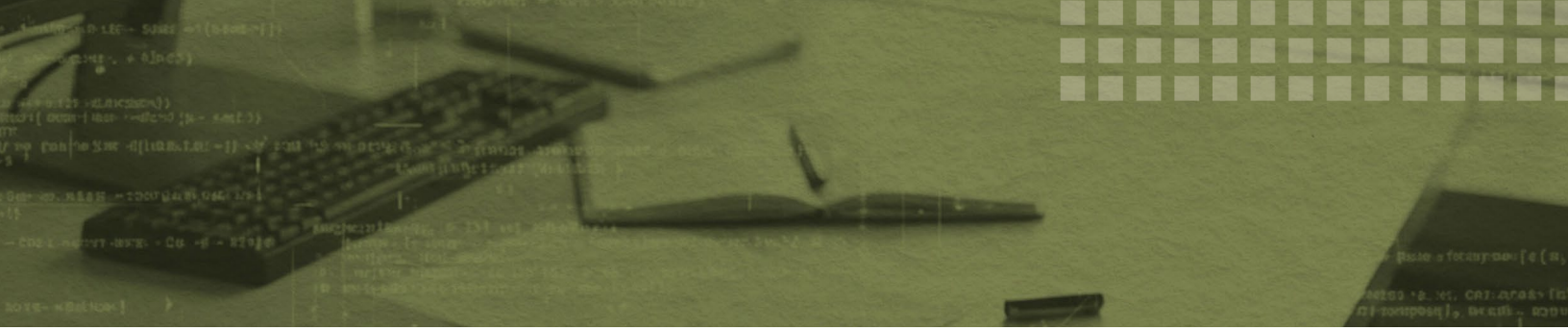
- **Dynamic Analysis**
  Authentication , Traffic Analysis, Input Validation, Privilege Escalation
- **Static Analysis**
  Inventory Files and Functionality, Missing Assembly Controls (e.g. DEP, ASLR, CFG, etc.), Database Audit

Common Attack Vectors

- File system
- The registry
- System memory
- Network communications
- Graphical user interfaces

Recommended Frequency

- **Annually**
- After **major updates**



Application Pentesting Considerations

Virtual Applications

A virtual application depends on where it is hosted, internally or in a virtualized environment, specifically referring to applications published through virtualization platforms such as Citrix and VMware. In other words, virtual applications are software applications that are delivered to users through virtualization technologies rather than being installed directly on their local machines.

Core Objective
Virtualized application environments introduce unique attack surfaces through hypervisor interactions, container isolation mechanisms, and shared resource architectures. Pentesting identifies and remediates vulnerabilities in virtualized deployments before attackers can exploit them to achieve container escape, compromise hypervisor integrity, perform lateral movement across virtual environments, or gain unauthorized access to co-located systems and data.

Focus Areas

- Policy and Sandbox Validation
- Data Exfiltration Channels
- Environment Enumeration and Network Segmentation
- Application Testing

Common Attack Vectors

- Sandbox escape
- Isolation bypass

Recommended Frequency

- **Semi-annually**
- **After significant changes** to the virtual environment

How to Action Your Pentest Insights

Once the lift of conducting a test is complete, the real fun starts. It's time to put your report into action. At a minimum, you'll walk away with a PDF that details the test scope, findings, and immediate next steps to fix vulnerabilities. But if your pentesting vendor only gives you a PDF, it's time for an upgrade.

NetSPI tracks the lifecycle of pentesting engagements in our platform from start to finish so you can see all of your information in one user-friendly interface. (We even have dark mode, too.) Access to a platform and free capabilities, such as attack surface visibility and detective controls testing, turns pentesting from a one-and-done project into an ongoing process of improving your overall security.

With your report on-screen in a sleek dashboard, you're ready to activate your pentest findings.

Step 1: Prioritize Risks

Many different strategies exist to help prioritize risks in a way that's best suited for your business. For example, many security teams rely on the CVSS as a baseline for criticality of vulnerabilities. In addition, teams may layer on prioritization based on a finding's ability to affect your business-critical operations. The best prioritization strategies combine these methods and more to create a system that effectively communicates the severity of a vulnerability to your business. No matter how you calculate risk, the first step after receiving a pentest report is to prioritize the most important findings.

Step 2: Create an Action Plan

Clear steps, assignees, deadlines, and KPIs ensure your findings are acted on. While remediating vulnerabilities may not be within your team's wheelhouse, our in-house consultants provide detailed remediation guidance to equip your team with the instructions to fix weaknesses and grow their skill sets.

Step 3: Build Security into Culture

No one likes making the same mistake twice. Use your report as a chance to level up your development team's approach to secure coding by sharing the findings with them. This will lessen the chance of the same vulnerabilities resurfacing in the future.

How to Mobilize Results from Pentest Reports *(cont.)*

Step 4: Communicate with Executives

Bridge the gap between the C-suite and your security team by sharing a summary of the report with executives. Remember to focus on the risk narrative and use the data and metrics to provide evidence.

Take a look at our resources on sharing pentest results with executives for more:

- Article: **From Pentest Report to Boardroom Strategy in 5 Steps**
- Podcast: **Translating Security for Your C-Suite**

Step 5: Validate Remediation

Could you imagine going through all this effort, only to implement a fix improperly, resulting in the same finding on your next test? We can't either. That's why NetSPI offers on-demand remediation testing so you can validate that you have closed the gap. NetSPI customers can request retesting through our platform, giving you full visibility into the status of your engagement at all times.

Step 6: Leverage Insights Long Term

Over time, you may start to notice trends in your reports, such as how your attack surface evolves, or the effectiveness of your detective controls. Use trends like these to guide your future security investments and support your long-term goals.

By following these steps, you'll take your report from plan to process and help your organization gain the most value from pentesting.

Ready to Build Your Application Pentesting Program?

Creating and optimizing an application pentesting program is complex. Stay focused on incremental improvements that ladder up to a more secure application ecosystem over time. A strong program is aligned to risks specific to your business with the flexibility to evolve as needed.

Pentesting is required for compliance in many cases, but its true value goes beyond checking the box. An insecure application can make or break a company's reputation and bottom line. As a security professional, your ultimate goal is to reduce the risk of successful attacks on your business. This makes proactively pentesting applications a strategic investment in the long-term health and success of the business.

When planning for your next test, trust NetSPI for our proprietary platform that streamlines testing. We identify, validate, and provide remediation guidance through a combination of automated and manual processes. Our proven methodology squeezes the value from each test, resulting in better overall security and stronger development practices for your team.

When you're planning your next application pentest, NetSPI's 350+ in-house security experts are ready to deliver.

Contact NetSPI tody to build your application pentesting programs