Security Audit Services

Q2 2026

Contents

1. Introduction

Financial Strides, in partnership with DataArt, provides a full range of solutions to help companies secure their products and infrastructure through a structured approach and consistent methodology based on industry-wide best practices and accompanying resources, such as OSSTMM, OWASP, WASC. We help our clients identify security issues, suggest remediation solutions and provide ongoing support to the customer's technical team.

The goal of this document is to describe typical tasks and methodology that Financial Strides and the DataArt Security Team follow during security audits, including:

- Black Box external network penetration test
- Internal network penetration test
- Grey Box web application penetration test
- Grey Box native application penetration test
- White Box security code review
- White Box cloud audit infrastructure review

Please refer to the Approach and Methodology section for a detailed description of these activities and utilized tools.

2. Approach and Methodology

2.1. External Network Penetration Test (Black Box)

The external network penetration test focuses on systems and services accessible from the Internet and aims to determine the extent to which these systems can be breached by an attacker with no insider knowledge but possessing skill and motivation.

Methodology

Our methodology for external network penetration (Black Box) tests consists of several phases:

Network Mapping

During the network mapping phase, we run several scans to create a network map of the target environment. This effort also identifies accessible systems, ports, services, and pinpoints other potential entry points an attacker might target.

System Classification

This phase involves classification of discovered systems using various fingerprinting methods. After each system is classified, the network map is updated to reflect each system's function and operating system.

Vulnerability Discovery

During the vulnerability discovery phase, we analyze all potentially exploitable attack vectors. During this step, we use a large working knowledge of exploit techniques, public information and the results of private vulnerability research.

The list below illustrates some of different vulnerability classes that we cover during that phase. The list is not intended to be exhaustive and the actual testing performed depends on the specifics of the organization being tested.

- Visible IP services
- Use of insecure protocols
- Misconfigured firewall rules
- Known OS/application flaws
- Missing security patches
- Weak configuration
- Web server vulnerabilities
- TLS weaknesses
- Remote code execution
- Buffer overflows
- Information disclosure

Exploitation

During the exploitation phase, our team leverages the identified vulnerabilities to launch its own attacks against the targeted systems. The attacks aim to obtain access to restricted data, take control of systems, impersonate users and perform other actions designed to demonstrate the potential consequences of the vulnerabilities discovered. During this phase, we also attempt to chain exploits to further penetrate the targeted systems, escalate access privileges and identify additional vulnerabilities.

Reporting

During reporting phase, we create a detailed report which includes an overview of findings and suggested remediation steps. The report will include the following sections:

- Executive summary
- Penetration testing methodology
- Scope of the penetration test
- Prioritized overview of findings
- Recommendations for improvement
- Supporting data and figures

Tools

- Nmap
- Nessus Professional
- Metasploit
- Nikto
- Fiddler
- SSLScan
- Wireshark
- Nexpose
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)
- Any other tools as necessary

2.2. Internal Network Penetration Test

During the internal penetration test, we emulate a malicious employee and attempt gaining an unauthorized access to organization data and resources. The test indicates if company efficiently prevents users or attackers who gained user level access from elevating their privileges or compromising organizational security in other ways.

Methodology

The methodology of the internal penetration test is similar to the external network penetration test; however, the attacker has access to the company local network and credentials of a non-privileged employee.

The list below illustrates some additional L2/L3 vulnerability classes that we cover during internal penetration tests. The list is not intended to be exhaustive and the actual testing performed depends on the specifics of the organization being tested.

- VLAN hopping
- ARP cache poisoning
- IP redirection
- DHCP weaknesses
- Protocol fuzzing
- Session hijacking/replay
- Password capture
- Insufficient access control
- Insufficient data protection
- Buffer overflows
- Abuse of functionality
- Information disclosure

Tools

- Nmap
- Nessus Professional
- Metasploit
- Nikto
- Wireshark
- Nexpose
- Kali Linux distribution
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)
- Other tools as necessary

2.3. Web Application Penetration Test (Grey Box)

The purpose of this test is to determine whether an attacker could compromise corporate web applications to get unauthorized access to company resources or to data of other users.

During the application penetration test, we mimic an external attacker without prior knowledge of the environment, or an attacker with user-level access within the application. We attempt to bypass security controls by taking advantage of the discovered vulnerabilities using automated and manual techniques.

Methodology

Our application penetration testing projects involve a methodology comprised of the following five phases: planning of project activities, information gathering, vulnerability discovery, vulnerability exploitation, and reporting. The process is designed to identify security issues and vulnerabilities in targeted applications, and to provide clients with a clear vision of the security state of their application as well as potential ways on how it could be compromised by real attackers.

Planning

As the initial step, Financial Strides defines and documents assessment objectives, scope, and rules of engagement. Financial Strides and DataArt conduct interviews with stakeholders to gain a thorough understanding of the client's goals and needs, security and compliance requirements, business risks, scope of the assessment and other related factors.

Information Gathering

During the information gathering phase, DataArt test engineers collect and examine key information about the application and its infrastructure: application functionality, use cases, user roles, architecture, security mechanisms, security-critical areas, hosting environment, and more. That information allows test engineers to properly target automated scanning software, better focus the manual testing process and investigate possible attack vectors.

Vulnerability Discovery

During the vulnerability discovery phase, our team uses both automated tools and manual techniques to survey the targeted environment and thoroughly identify application vulnerabilities. Automated tools are used initially to enumerate application resources and pick the most common issues. After that, heavy manual testing identifies any remaining issues which are often overlooked by many competitors.

The manual testing focuses on vulnerabilities including but not limited to:

Authentication and Access Control	• Weak password policy • Brute-force login • Insecure credentials transmission	• Password recovery weaknesses • SSO implementation flaws • Broken access controls
Session Management	• Insecure token generation	• Session hijacking • Session fixation

	• Insecure token transmission • Cookie poisoning	• Insecure session termination
Command Injection Flaws	• SQL injection • Xpath injection • OS command injection • Code injection	• Path manipulation • Buffer overflow • Cross-site scripting
Client-Side Technology Flaws	• Reliance on client-side validation • Insecure local data storage • Same-origin policy	• AJAX/web service flaws • Java/ActiveX/Flash/Silverlight objects
Application Logic Flaws	• Privilege escalation • Sensitive information disclosure • File I/O implementation defects • Insecure use of cryptography	• Cross-site request forgery • Weak data validation • Race conditions • CPU intensive functions
Information Disclosure	• Server banners • Verbose error messages • Client-side code comments • Source code disclosure	• Accounts enumeration • Cached web content • Local privacy issues
Platform Misconfiguration	• Default content • Default administrative credentials	• Improper web server configuration • SSL and transport layer weaknesses
LLM Vulnerabilities	• Prompt Injections • Insecure output handling • Supply chain • Permission issues	• Sensitive information disclosure • Excessive agency • Insecure plugins and agents

The following security standards will be referenced as a part of this assessment: OWASP Top 10, OWASP API Security Top 10, OWASP TOP 10 LLM

Vulnerability Exploitation

During this phase, any potential vulnerability found is manually investigated, researched, and an attempt is made to exploit. In exploiting vulnerability, we will make an attempt to either

gain unauthorized access to the target system or extract sensitive data from it. An exploit is considered successful if we were able to achieve either of these objectives.

For applications leveraging Large Language Models, we conduct comprehensive security assessments aimed at identifying potential risks that could lead to sensitive data exposure or unauthorized system access. Our experts evaluate critical areas such as input manipulation, prompt injection, and plugin vulnerabilities to uncover weaknesses in the LLM's design, configuration, and integration with external systems.

Each system that is compromised will be examined for the existence of critical data and files. If we find such data to be accessible, a sample of this data will be downloaded from the system and securely stored by us until the presentation of deliverables.

As systems or applications are compromised, our customer's key security contacts will be notified. At that time, the customer's contacts will be given the opportunity to decide if the particular system should undergo additional tests. If they decide to have us continue, additional techniques will be used to further penetrate the target system and the environment as a whole. This can include installation of network sniffers, remote management tools, connectivity tools etc.

Reporting

Following the completion of the security assessment project, we deliver a detailed report based on the team's findings. For each finding, we also include a close overview, evidence (screenshots), analysis of potential impact, and actionable recommendations for remediation or mitigation.

Tools

- Vulnerability Scanners (Nessus Professional, ActiveScan++, BurpBountyPro)
- Security Testing Platforms (Burp Suite Pro, OWASP Zap)
- Enumeration tools (dirsearch, gobuster, Wfuzz, findomain)
- Injection testing tools (SQLmap, XSSStrike, XSSHunter)
- Brute force tools (hashcat, john, hydra, WebSocket BF)
- Exploitation tools (phpgcc, ruby marshaling, ysoserial, graphql-voyager, JWT tool)
- CMS security testing tools (wpscan, droopscan, joomscan)
- Payload and shells repositories (SecLists, PayloadsAllTheThings)
- SSL/TLS Analysis Tools (sslscan, testssl)
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)
- Other tools as necessary

2.4. Native Application Penetration Test

Such a test is aimed at the same target, as web application penetration test – identify whether an attacker could compromise native desktop or mobile applications to get unauthorized access to personal or corporate information, company resources or to data of other users. However, the focus is shifted to breaking local privacy and bypassing platform-specific APIs and mechanisms used for data protection.

When performing penetration testing of native applications, we mimic an external attacker without prior knowledge of the environment, or an attacker with physical access to a victim's device and user-level access within the application. We will attempt to bypass both application and platform security controls by taking advantage of the discovered vulnerabilities using automated and manual techniques.

Methodology

Our approach to penetration testing of native applications consists of the same major steps described in the previous section planning, information gathering, vulnerability discovery, vulnerability exploitation, and reporting, – but the techniques are different. During vulnerability discovery phase, we focus on reverse-engineering application logic and its security controls, dynamic application analysis and inspection of data which is stored locally. We analyze all communications of the application with remote services and ensure security of any transmitted data. During the test, we look at the application from an attacker's viewpoint and try to devise and launch attacks.

Native application testing focuses on possible vulnerabilities in mobile application logic, looking for issues including but not limited to:

- Local data storage
- Caching and temporary files
- Logging
- Privacy issues
- Information leakage
- WebView vulnerabilities
- SSL and transport layer weaknesses
- Authentication and session management defects
- Unmanaged code and memory access
- Memory leaks

We also uncover server-side APIs used by the applications, and these APIs undergo a Web Application penetration testing, as described in the previous section. If native applications expose various widgets or use 3rd party external modules, like static or dynamic libraries, they're tested as well, and the issues found there are included into the final report, too.

Tools

- Proxies (Burp Suite Pro, OWASP ZAP)
- Platform-specific reverse engineering tools (e.g., Frida, Objection for iOS)
- Toolchains and SDKs for native and hybrid apps (e.g., gcc, clang and iOS SDK)
- Tools for extracting secrets and passwords (e.g., chainbreaker for macOS)
- Debuggers and runtime modifiers (e.g., gdb, cycript)
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)
- Other tools as necessary

2.5. Security Code Review (White Box Testing)

A security code review is a "white box" testing activity aimed at analyzing application source code and determining potential security weaknesses and flaws in the code.

Methodology

Our security code review projects include both static and manual analysis.

Static Analysis

Our team uses several static analysis tools to survey the source code of the targeted applications and identify potential vulnerabilities. These tools are capable of identifying the following types of issues:

- Weak cryptographic algorithms and hash functions
- Untrusted input parameters
- Potential SQL/HQL Injection
- Potential LDAP injection
- Potential path traversal
- Potential command injection
- Potential XPath injection
- Potential HTTP response splitting
- Potential template injection
- Potential XSS
- Unvalidated redirects
- Potentially sensitive data in a cookie
- Cookie without HttpOnly and Secure flags
- Insufficient SSL/TLS server authentication
- Trust boundary violations
- Hardcoded keys and passwords

Manual Code Review

For manual reviews, our team adopts the process and guidelines described in the OWASP Code Review Guide V1.1. Additionally, we reference the list of application security requirements from the OWASP Application Security Verification Standard 3.0 (level 3).

Our team will perform the following actions as part of this task:

- Get familiar with the business purpose of the applications
- Identify different types of threat agents and potential attack vectors
- Identify all application inputs and outputs
- Perform dynamic and static data flow analysis
- Perform analysis of application transactions
- Review implementation of application security controls
- Crawl the source code for specific security vulnerabilities

The manual code review covers the most critical security controls and vulnerability areas such:

- Input handling
- Data validation
- Authentication
- Session management
- Authorization
- Cryptography
- Error handling
- Logging
- Security configuration
- Application logic defects
- Concurrency issues
- Application-level denial-of-service

Tools

- SAST (SonarQube, Semgrep, and language-specific tools)
- SCA and Container Scanning (Snyk, Trivy)
- Secret Scanning (GitLeaks, TruffleHog)
- IaC Scanning (Trivy, Checkov)
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)

2.6. Cloud Audit (White Box Infrastructure Review)

Cloud audit is a "white box" infrastructure review and testing activity which is focused on the following goals:

- Check that the cloud infrastructure and security controls are implemented in line with client's security policies and common best practices
- Find non-addressed security gaps and identify the related technical issues within cloud environments and suggest the improvements

Methodology

Our cloud assessment framework consists of four phases: information gathering, interviews, manual and automated assessment, reporting.

Information Gathering

We introduce our assessors to the client's cloud maintenance team and asks the team members to provide security-related documentation and fill in a special questionnaire which covers the key aspects of cloud security. The questionnaire includes such aspects as authentication, access control and management, audit, application and network level security, maintenance, and other. The list of required artifacts includes, but is not limited to:

- Network and architecture diagrams
- Application catalogue
- Results of previous vulnerability scans and reports from security tools
- Various configurations (applications, network, etc.)
- Samples of application logs and audit trails

Once the artifacts and the filled questionnaire are shared with the assessment team, we review and analyses them to identify the architecture of the cloud, types of deployed services, security controls used, as well as the key people and processes.

Interviews

Following the completion of the previous phase, we arrange a number of interviews with the key people responsible for the cloud: network engineers, technical architects, leads of support and maintenance teams, security and compliance officers. Based on the gathered information, the assessors prepare specific questions for each team member and ask for quick demos and screen sharing sessions, if possible.

If there are some blind spots discovered during the assessment, we arrange an extra round of interviews to clarify uncertain items, ask extra questions and gain access to additional materials.

Manual and Automated Assessment

Our team performs automated assessment of the cloud infrastructure to validate the existing security controls using pre-defined checklists, find out potential problems in their configurations and translate them into reports. The assessment team also inspects and analyses the whole setup manually, either via web console or cloud-provided APIs, and collects the evidences that the security controls work as described in policies and best practices. All deviations from recommended configurations are recorded as well. The following things are usually examined:

- Trust boundaries
- User authentication and access control
- Separation of roles and duties
- Data protection measures in transit and at rest
- Secure remote and administrative access
- Attack detection and response mechanisms
- Secure backups and disaster recovery plan

Reporting

At the final step of the audit, our team collaborates with the cloud team on disputable findings, defines the criticality of each item and creates an audit report which includes the executive summary section, description of methodology, definition of audit scope and prioritized overview of issues. Each finding is accommodated with criticality degree, description, evidences (screenshots, links to configurations, video and voice recordings, etc.) and recommendations on remediation. This report is shared with the team and discussed on a follow-up meeting.

Tools

- Scout Suite, Prowler
- PMapper, CloudSploit
- Kali Linux
- AI Tools (Claude Enterprise, OpenAI Codex, Microsoft Copilot, where feasible and allowed by the client)
- Other tools as necessary

3. Deliverables

Following the completion of each task, we deliver a detailed report based on the team's findings. The reports include the following sections:

- Executive summary (in a non-technical fashion suitable for senior management)
- Assessment methodology
- Scope of the assessment
- Attack narrative (where applicable)
- Prioritized overview of findings
- Recommendations for improvement
- Supporting data and figures

All identified vulnerabilities are classified by their cause and then prioritized by an estimated risk rating. For each finding, we also include a close overview, evidence (screenshots), analysis of potential impact, and actionable recommendations for remediation or mitigation.

The customer is immediately alerted of any critical vulnerabilities found during audit so they may immediately take action without waiting for the full report.

In the event a retest is performed, issues are updated with retest notes that indicate the tests performed to validate proper fixes have or have not been properly implemented within the environment.

Criteria for Risk Rating

Assignment of risk ratings to identified vulnerabilities is one of the key elements of the security assessment. It allows creation of a prioritized list of findings so the client can start remediation from issues that pose the greatest risk to their organization.

Our security analysts perform an intelligent analysis to determine and assign a risk rating to each identified issue. The following information obtained during assessment project is considered in the calculation of a risk rating:

- Likelihood of attack
- Potential impact of exploitation
- Level of skill required to execute attack
- Number of instances of the vulnerability
- Mitigating controls

The table below outlines risks ratings:

Rating	Description
CRITICAL	Severe issues that can easily be exploited to immediately impact the environment.

HIGH	Major issues that can be exploited to impact the environment, however their applicability is limited, or additional effort is required for exploitation.
MEDIUM	Moderate security issues that require some effort to successfully impact the environment.
LOW	Security issues that have a limited or trivial impact to the environment.

4. Team Certifications

All members of the test team have one or more of the following certifications:

OSCP (Offensive Security Certified Professional)

CEH (Certified Ethical Hacker)

CPTS (Certified Penetration Testing Specialist)

CREST Registered Penetration Tester

BSCP (Burp Suite Certified Practitioner)

OSWE (Offensive Security Web Expert)

eCPPT (Certified Professional Penetration Tester)

eCPTXv2 (Certified Penetration Tester eXtreme v2)

eWPTXv2 (Web Application Penetration Tester eXtreme v2)

eMAPT (Mobile Penetration Tester)

5. Project Management

Financial Strides provides project management for its services to reduce the amount of administrative and other challenges present when conducting an assessment.

Role and responsibility of project manager is as follows:

- Planning and scoping
- Report on a daily basis to all parties in the project on all aspects of the project
- Establish and maintain secure communication among participants
- Manage the project timeline, budget and expectations
- Define and reach to service groups to respond to project needs
- Collect and update all status, preliminary and final reports
- Manage projects financial reporting including timesheets and invoices
- Review the status of all audit activities on weekly basis.

Engagement Overview

Our security audit projects involve the following phases:

- **Planning**: our experts work with the client to clearly define and document assessment objectives, scope, and rules of engagement. During that phase, the client provides necessary access credentials, technical details and other information required for the task. A kick-off meeting is held at the end of the phase to finalize engagement rules.
- **Assessment**: our team performs assessments according to the goals set during the planning phase. Some assistance is typically required from the client such as to resolve any technical issues, explain certain things, discuss any critical issues identified, etc.
- **Report presentation:** we deliver the audit reports and presents the overall results to the client stakeholders.
- **Remediation support:** we can provide ad-hoc support to the client engineers during issue remediation.
- **Re-testing**: If applicable, our teams can perform re-testing of identified issues and delivers a new revision of the report.

6. Appendix: DataArt Info

DataArt is a global software engineering firm that takes a uniquely human approach to solving problems. With over 20 years of experience, teams of highly-trained engineers around the world, deep industry sector knowledge, and ongoing technology research, we help clients create custom software that improves their operations and opens new markets.

Powered by our People First principle, we work with clients at any scale and on any platform, and adapt alongside them as they evolve.

We integrate our engineering excellence with deeply human values that drive our business and our approach to relationships: curiosity, empathy, trust, honesty, and intuition. These qualities help us deliver high-value, high-quality solutions that our clients depend on, and lifetime partnerships they believe in.

Global locations:

- New York
- London
- Zurich, Switzerland
- Munich, Germany
- Eastern Europe
- Latin America

Quick Facts about DataArt:

- Founded in 1997, transparent, privately held, profitable;
- Offering security audit testing since 2012;
- $378M revenue in 2022;
- 95% return clients;
- 4.7 average tenure;
- We partner with 400+ leading companies.
- Our top 20 clients have worked with us for 7+ years on average;
- 40+ global locations;
- 5,000+ Consultants & Engineers;
- 87 % employee retention in 2023;
- SOC 2 Type II certified, prioritizing security both internally and for our clients;
- No open litigations, legal history is limited to collections matters.

7. Appendix: Financial Strides Info

Financial Strides is a FinTech consultancy firm with a presence in California and the United Kingdom.

Financial Strides main areas of **product expertise** are:

- Online and Mobile Account Opening: Customer Identification Process ("CIP"), AML/BSA and OFAC ("KYC"), Fraud detection and Red Flags:
- Payment Protocols: Card payments protocols, Fast Payments: Push-to-Debit, RTP, Zelle, ACH
- Online and Mobile Account Servicing: Mobile-responsive browser-based, Native mobile applications for iOS and Android, Email and mobile push messaging management
- IT Set Up and Management: Managed Services outsourcing and supervision, Cloud based set-ups (AWS, Google, Azure), IT Architectures optimized for PCI-DSS and GLBA compliance, Business Continuity and Disaster Recovery Plans and processes

We assist Clients with establishing their **compliance framework** by:

- Evaluating their existing policies & processes
- Creating custom written policies & documented processes where needed
- Training staff for specific regulatory compliance: AML/BSA & OFAC, UDAAP, Privacy & GLBA, ID Theft & Red Flags, Regulation E

We perform **security & process control evaluations** through:

- Penetration testing: "black box," "grey box" as per OWASP and PCI standards
- PCI assistance with Self-Assessment Questionnaires
- SOC 2, ISO 27001 preparation work prior to formal audits