

الجمهورية الشعبية الديمقراطية الجزائرية
People's Democratic Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة العليا للإعلام الآلي 8 ماي 5491 - سيدي بلعباس
Higher School of Computer Science
8 Mai 1945 - Sidi Bel Abbès



Engineering Graduation Thesis

To obtain the diploma of **Engineering Degree**

Field of Study: **Computer Science**

Specialization: **Computer Systems Engineering (ISI)**

Theme

Pentest Automation

Presented by
Abderraouf Atsamnia
Ibrahim Senouci

Defended on: **June, 2026**
In front of the jury composed of

Dr. KHALDI Miloud
Dr. BABA-AHMED Manel
M MEGHNINE Islem
Dr. BENDELLA Mohamed Salih

President of the Jury
Thesis Supervisor
Co-Supervisor
Examiner

Academic Year: 2025/2026

Abstract

The growing complexity of digital infrastructures has significantly expanded the attack surface exposed to malicious actors, making systematic and efficient security assessment increasingly difficult to sustain. Penetration testing has established itself as a cornerstone practice for identifying vulnerabilities before they can be exploited, yet current practice remains affected by structural limitations: fragmented tool ecosystems, manual workflow orchestration, absence of structured lifecycle management, and insufficient governance mechanisms.

This thesis presents the design and incremental implementation of **PentestOPS**, a modular penetration testing automation platform that addresses these limitations through a unified, actor-driven architecture. The system organizes the penetration testing lifecycle around four primary actors, namely the Pentest Admin, the Pentester, the Client, and the Automation Engine, each assigned clearly delineated responsibilities and interaction boundaries. A formal requirements analysis, expressed through UML use case, sequence, and activity diagrams, guided the architectural decisions and ensured alignment between system behavior and real-world engagement workflows.

The platform is built on a modular monolithic architecture using Flask, PostgreSQL, and a vanilla JS single-page frontend communicating through a RESTful API. Its core contribution is a pipeline-based automation engine that orchestrates passive subdomain enumeration, DNS resolution, HTTP host probing, port scanning, web crawling, JavaScript analysis, and template-based vulnerability detection through controlled subprocess execution, structured output normalization, and semantic deduplication. A pre-engagement validation mechanism enforces mutual scope approval before any scanning activity can be initiated, and a RBAC model governs all platform operations.

A comparative analysis of eleven existing tools and platforms revealed a consistent structural divide between automation-oriented solutions lacking platform infrastructure and platform-oriented solutions lacking native pipeline execution. PentestOPS addresses this gap by natively combining both within a single, open, and extensible architecture.

The implemented system validates the proposed architectural approach and demonstrates that modular automation, structured workflow orchestration, and centralized lifecycle management can improve the efficiency, consistency, and traceability of modern penetration testing operations.

Keywords: Penetration Testing, Reconnaissance Automation, Workflow Orchestration, Modular Architecture, Attack Surface Discovery, Vulnerability Management, REST API, Role-Based Access Control.

الملخص

أدّى التوسّع المتزايد في البنى التحتية الرقمية إلى توسيع كبير لسطح الهجوم المعرّض للجهات الخبيثة، مما جعل إجراء تقييمات أمنية منهجية وفعّالة أمرًا بالغ الصعوبة. وقد رتّب اختبار الاختراق نفسه بوصفه ممارسةً محوريةً للكشف عن الثغرات قبل أن يتمكن المهاجمون من استغلالها، غير أن الواقع العملي الراهن لا يزال يعاني من قيود هيكلية جوهرية، تشمل: تشتت الأدوات المستخدمة، والتنسيق اليدوي للعمليات، وغياب إطار منظم لإدارة دورة حياة الاختبار، وضعف آليات الحوكمة.

تقدّم هذه الأطروحة تصميم وتطوير تدريجي لمنصة **PentestOPS**، وهي منصة أتمتة معيارية لاختبار الاختراق تعالج هذه القيود من خلال بنية موحدة قائمة على نموذج الأطراف الفاعلة. ينظّم النظام دورة حياة اختبار الاختراق حول أربعة أطراف رئيسية، وهي: مسؤول اختبار الاختراق، والمختبر، والعميل، ومحرك الأتمتة، إذ يُحدّد لكلٍ منها نطاق مسؤولياتها وصلاحياتها وحدود تفاعلها بشكل صريح. وقد اعتمد التحليل التفصيلي للمتطلبات على مخططات UML من نوع حالات الاستخدام والتسلسل والنشاط، مما وفر أساساً هندسياً متيناً يضمن توافق سلوك النظام مع سير العمل الفعلي في بيئات اختبار الاختراق.

تعتمد المنصة على بنية أحادية معيارية مبنية باستخدام إطار العمل Flask وقاعدة البيانات PostgreSQL وواجهة أمامية أحادية الصفحة مكتوبة بـ JavaScript الأصلية، تتواصل مع الخلفية عبر API بمعيار REST. ويتمثّل الإسهام الجوهرى للمنصة في محرك أتمتة قائم على خطوط المعالجة، يُنسّق تنفيذ عدة مراحل متتالية، تشمل: الاستطلاع السلبي لأسماء النطاقات الفرعية، وتحليل DNS، واستطلاع المضيفين عبر HTTP، ومسح المنافذ، وزحف الويب، وتحليل ملفات JavaScript والكشف عن الثغرات بالقوالب؛ وذلك من خلال تنفيذ محكوم للأدوات الخارجية، وتوحيد المخرجات، وإزالة التكرارات الدلالية. كما يُطبّق نظام التحقق من مرحلة ما قبل المشاركة لضمان الموافقة المتبادلة على نطاق العمل قبل الشروع في أي نشاط مسحي، فيما يُشرف نظام التحكم في الوصول المبني على الأدوار RBAC على جميع عمليات المنصة.

كشفت التحليل المقارن لأحد عشر أداة ومنصة موجودة عن فجوة هيكلية واضحة بين حلول الأتمتة التي تفتقر إلى بنية المنصة من جهة، وحلول المنصات التي تخلو من آليات تنفيذ خطوط المعالجة الأصلية من جهة أخرى. وتسدّ منصة PentestOPS هذه الفجوة بالجمع الأصيل بين المقاربتين ضمن بنية واحدة مفتوحة وقابلة للتوسّع.

يُثبت النظام المنجز صحّة المقاربة المعمارية المقترحة، ويبيّن أن الأتمتة المعيارية والتنسيق المنظم لسير العمل وإدارة دورة الحياة المركزية قادرة على تحسين كفاءة عمليات اختبار الاختراق الحديث وآساقها وقابليتها للتبع.

الكلمات المفتاحية: اختبار الاختراق، أتمتة الاستطلاع، تنسيق سير العمل، البنية المعيارية، اكتشاف سطح الهجوم، إدارة الثغرات، واجهة برمجة التطبيقات، التحكم في الوصول المبني على الأدوار

Résumé

La complexité croissante des infrastructures numériques a considérablement élargi la surface d'attaque exposée aux acteurs malveillants, rendant l'évaluation systématique et efficace de la sécurité de plus en plus difficile à maintenir. Les tests d'intrusion se sont imposés comme une pratique incontournable pour identifier les vulnérabilités avant qu'elles ne puissent être exploitées, mais la pratique actuelle reste affectée par des limitations structurelles majeures, notamment la fragmentation des écosystèmes d'outils, l'orchestration manuelle des flux de travail, l'absence de gestion structurée du cycle de vie des engagements, et l'insuffisance des mécanismes de gouvernance.

Cette thèse présente la conception et l'implémentation incrémentale de **PentestOPS**, une plateforme d'automatisation modulaire des tests d'intrusion qui répond à ces limitations à travers une architecture unifiée pilotée par les acteurs. Le système organise le cycle de vie des tests d'intrusion autour de quatre acteurs principaux, à savoir l'administrateur de pentest, le testeur d'intrusion, le client et le moteur d'automatisation, chacun disposant de responsabilités, de permissions et de frontières d'interaction clairement définies. Une analyse formelle des exigences, exprimée à travers des diagrammes UML de cas d'utilisation, de séquence et d'activité, a guidé les décisions architecturales et assuré l'alignement entre le comportement du système et les flux de travail réels des engagements de sécurité.

La plateforme repose sur une architecture monolithique modulaire utilisant Flask, PostgreSQL et une interface frontale monopage en JavaScript natif, communiquant avec le backend via une API RESTful. Sa contribution centrale est un moteur d'automatisation fondé sur des pipelines de traitement, qui orchestre l'énumération passive de sous-domaines, la résolution DNS, le sondage HTTP des hôtes, le balayage de ports, l'exploration web, l'analyse JavaScript et la détection de vulnérabilités par gabarits, à travers une exécution contrôlée des outils externes, une normalisation des sorties et une déduplication sémantique. Un mécanisme de validation de pré-engagement impose une approbation mutuelle de la portée avant toute activité de balayage, tandis qu'un modèle de contrôle d'accès basé sur les rôles RBAC gouverne l'ensemble des opérations de la plateforme.

Une analyse comparative de onze outils et plateformes existants a révélé une division structurelle persistante entre les solutions orientées automatisation, qui manquent d'infrastructure de plateforme, et les solutions orientées plateforme, qui ne disposent pas d'un pipeline d'exécution natif. PentestOPS comble cette lacune en combinant nativement les deux approches au sein d'une architecture unique, ouverte et extensible.

Le système implémenté valide l'approche architecturale proposée et démontre que l'automatisation modulaire, l'orchestration structurée des flux de travail et la gestion centralisée du cycle de vie peuvent améliorer l'efficacité, la cohérence et la traçabilité des opérations modernes de test d'intrusion.

Mots-clés : Test d'intrusion, Automatisation de la reconnaissance, Orchestration de flux de travail, Architecture modulaire, Découverte de la surface d'attaque, Gestion des vulnérabilités, API REST, Contrôle d'accès basé sur les rôles