International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

AI-Powered Penetration Testing Agent

G. Prabakaran, Kaniga M, Kanimozhi M, Keerthana K, Minisha M

Assistant Professor, Dept. of Information Technology, Vivekanandha College of Technology for Women, Tiruchengode, Tamil Nadu, India

B. Tech, Dept. of Information Technology, Vivekanandha College of Technology for Women, Tiruchengode, Tamil Nadu, India

B. Tech, Dept. of Information Technology, Vivekanandha College of Technology for Women, Tiruchengode, Tamil Nadu, India

B. Tech, Dept. of Information Technology, Vivekanandha College of Technology for Women, Tiruchengode, Tamil Nadu, India

B. Tech, Dept. of Information Technology, Vivekanandha College of Technology for Women, Tiruchengode, Tamil Nadu, India

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: Penetration testing has long demanded deep expertise and a working familiarity with a sprawling toolkit — tools that rarely talk to each other. ScanAI is built to change that. It is an autonomous, AI-driven security assessment agent that lets practitioners describe what they want in plain English and then takes care of the rest: picking the right scanners, tuning their configurations, chaining them in the correct order, pulling together the findings, spotting the attack paths that matter, and producing a report that is actually actionable. At its core, ScanAI runs on Google Gemini and brings together 23 specialised scanner modules governed by 106 YAML workflow profiles, collectively covering network infrastructure, web application security, open-source intelligence, and secrets detection. A dedicated Smart Correlation Engine ties all of this together — synthesising outputs across modules, computing composite risk scores on a 0–100 scale, and mapping out remediation priorities. When tested across eight attack scenario categories, ScanAI returned an average precision of 96.6%, recall of 95.8%, and F1-score of 96.1%. Head-to-head against tools like Burp Suite, OWASP ZAP, and Nessus, it pulls ahead on automation depth, scanner coverage, and the kind of cross-domain intelligence that none of them attempt.

KEYWORDS: Autonomous penetration testing, AI-driven security assessment, Natural language security orchestration, Multi-tool vulnerability scanning, Smart correlation engine, Attack path analysis, Automated remediation prioritization

I. INTRODUCTION

Anyone who has run a real penetration test knows how much of the work is not about hacking — it's about managing tools. You fire up Nmap, comb through ports, switch to Nikto, juggle output files, cross-reference CVEs manually, and somehow try to weave all of that into a coherent picture of what is actually exploitable. For seasoned professionals this is second nature, but it is also slow, inconsistent, and hard to scale. Organisations need security assessments more frequently than ever, while the pool of people qualified to run them has not kept pace with demand.

Recent advances in large language models have opened a genuinely different way to approach this problem. Models like Google Gemini can parse intent, reason over structured data, and produce well-formed outputs — capabilities that map surprisingly well onto what a penetration tester does when planning an engagement. The question is whether those capabilities can be wired up to real tools in a way that holds together under operational conditions.

ScanAI is our answer to that question. Rather than wrapping a chatbot around a single scanner, it functions as a full orchestration layer: accepting a plain-English objective, mapping it to the right combination of tools, running them in sequence, correlating what comes back, and generating a structured report — all without the operator needing to touch

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

a command-line flag. The goal is not to replace the experienced security professional but to give them a force multiplier, and to put meaningful assessment capability within reach of teams that currently lack it.

The contributions of this work are summarised below:

Design and implementation of ScanAI — an autonomous penetration testing agent that integrates Google Gemini with 23 specialised security scanner modules covering four distinct assessment domains.

A YAML-based workflow system comprising 106 fine-grained scan profiles, enabling precise and customisable security assessments through natural language commands alone.

A Smart Correlation Engine capable of performing cross-scanner synthesis, identifying attack paths, computing composite risk scores, and generating ranked remediation guidance.

Nine pre-built multi-scanner attack chains covering full reconnaissance, web application assessment, phishing analysis, secrets detection, and internal network assessment.

Comprehensive evaluation demonstrating 96.6% average precision and 96.1% F1-score across eight distinct attack scenario categories.

II. RELATED WORK

Established tools such as Nessus , OpenVAS, and OWASP ZAP have been the industry standard for automated vulnerability detection for years. They are reliable and well-understood, but they share a common limitation: they operate in isolation, offer no cross-domain intelligence, and expect the operator to handle orchestration manually. Burp Suite, the dominant tool for web application testing, requires considerable configuration and expertise before it becomes genuinely useful, and produces no synthesis across other scanner categories.

The use of machine learning to improve vulnerability detection has attracted sustained research interest. Ghaffarian and Shahriari published an influential survey of ML-based software vulnerability analysis methods, while transformer-based models such as CodeBERT have shown real promise for source-code-level vulnerability identification. These efforts have mostly remained at the detection layer, however, without addressing how findings from multiple tools should be orchestrated and interpreted together.The emergence of GPT-class language models prompted a new line of inquiry. PentestGPT demonstrated that an LLM can meaningfully guide a penetration tester through an engagement using natural conversation — a genuine step forward, though the human operator still needs to manually execute every suggested command. ScanAI takes that further: the model does not just advise, it acts. Tool selection, execution, and correlation happen autonomously, without the operator as an intermediary.Automated OSINT collection and exploit path generation have each been studied as standalone problems. Wang et al. proposed a multi-agent framework for penetration testing using LLMs, introducing modular agent cooperation as an architecture worth exploring. What has remained missing is a single, production-ready framework that brings natural language interaction, multi-tool orchestration, secrets detection, and cross-domain correlation together in one place. ScanAI is designed to fill that gap.

III. SYSTEM ARCHITECTURE

ScanAI is built as a layered autonomous agent system. Six principal components work together in sequence: a Natural Language Interface, an AI Query Interpreter, a Workflow Engine, Scanner Modules, a Smart Correlation Engine, and an AI Report Generator. Data moves through the pipeline in a single direction:

User Prompt → AI Query Interpreter → Workflow Engine → Scanner Modules (23) → Smart Correlation Engine → AI Report Generator + Session Persistence

A. Natural Language Interface

The entry point is a terminal interface built with the Rich Python library. Operators type what they want in plain English — there are no flags to memorise, no syntax to look up. Commands like "full recon example.com", "find secrets in github.com/org/repo", or "web attack https://target.com" are all that is needed to kick off a multi-tool assessment. This abstraction is not cosmetic: it genuinely removes the knowledge overhead that has historically made comprehensive penetration testing inaccessible to many teams.

B. AI Query Interpreter (Google Gemini)

Behind the interface sits the AI Query Interpreter, powered by Google Gemini via LangChain. Its job is threefold: determine what the user actually wants, decide which of the 23 scanner modules are relevant, and select the most

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

appropriate workflow profile from the 106 available YAML definitions. That last step is where context matters — Gemini weighs factors like how stealthy the scan needs to be, what kind of target is involved, and how deep the assessment should go. The output is not a suggestion; it is a fully structured execution plan passed directly to the Workflow Engine.

C. Workflow Engine and YAML Profiles

The Workflow Engine takes the structured plan and runs it. Every scanner is governed by a YAML file that defines multiple profiles — each specifying a command template, execution timeout, and descriptive tags. Nmap, for instance, has ten profiles ranging from stealth_scan (-sS -T2 -f) to aggressive_scan (-A -T4) to vuln_scripts (--script vuln -sV). The engine also handles auto-chaining: if Nmap discovers open ports, it can automatically trigger Nuclei CVE template matching on those services, without waiting for human input.

D. Scanner Modules (23 Modules)

ScanAI currently integrates 23 scanner modules across four operational domains:

Network and Infrastructure: Nmap (10 profiles), DNS reconnaissance, subdomain enumeration, WHOIS intelligence gathering, SSL/TLS analysis, IP geolocation with Shodan enrichment, and SMB/NetBIOS enumeration.

Web Application: Nikto, Gobuster directory brute-forcing, Katana web crawling, Dalfox XSS scanning, SQLMap injection detection, Nuclei CVE template scanning, WhatWeb technology fingerprinting, security header auditing, WAF detection, and WPScan WordPress auditing.

OSINT and Intelligence: theHarvester email and subdomain harvesting, VirusTotal reputation analysis, URLScan screenshot capture, CVE database lookup, and Wayback Machine archive retrieval.

Secrets and Credentials: Titus, a high-performance secrets scanner running 459 detection rules that cover API keys, tokens, credentials, and private keys — with live validation of discovered credentials.

E. Smart Correlation Engine

Individual scanner outputs tell you what was found. The Smart Correlation Engine tells you what it means. After each scan session, it ingests structured results from every active module and looks for connections: an open port combined with a known CVE and an absent WAF, for example, suggests a plausible service exploitation path. The engine currently detects five primary attack path types — Service Exploitation, Unprotected Web Applications, WordPress CMS Exploitation, Information Disclosure, and Historical URL Exposure — and computes a single composite risk score on a 0–100 scale.

F. AI Report Generator and Session Persistence

When an assessment session closes, Google Gemini synthesises all correlated findings into a structured penetration testing report delivered in both Markdown and HTML. Each report includes an executive summary, per-scanner findings, attack path descriptions, the composite risk score, and a ranked list of remediation recommendations ordered by severity. The Session Persistence module serialises the full scan state so that long or complex assessments can be saved, resumed, and built upon across multiple work sessions.

IV. ATTACK CHAIN ORCHESTRATION

Nine pre-configured multi-scanner attack chains allow a single natural language prompt to trigger a coordinated, multi-stage assessment sequence. The full_recon chain — launched with something as simple as "full recon example.com" — runs eleven scanners in order: WHOIS, DNS enumeration, subdomain discovery, theHarvester OSINT collection, IP geolocation, Wayback Machine archive lookup, Nmap port scanning, WAF detection, WhatWeb fingerprinting, security header analysis, and SSL/TLS inspection. What would traditionally take an expert half a day to execute and correlate across eleven separate tools runs as a single, unattended operation.

Other chains follow the same logic. The web_attack chain sequences WAF detection, Nikto scanning, Katana crawling, Gobuster directory brute-forcing, Dalfox XSS scanning, SQLMap injection testing, and Nuclei CVE scanning into a complete web application assessment. The phishing_analysis chain combines VirusTotal reputation lookup, URLScan analysis, WHOIS domain age verification, and SSL certificate validation to deliver a rapid phishing verdict. The remaining chains handle vulnerability assessment, stealth reconnaissance, WordPress auditing, OSINT gathering, and internal network penetration testing.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

V. EXPERIMENTAL EVALUATION

A. Experimental Setup

All testing was conducted on a Kali Linux environment running Python 3.10, connected to a 1 Gbps dedicated test network segment. Target environments included DVWA (Damn Vulnerable Web Application), Metasploitable 2, a WordPress installation with known vulnerable plugins, and simulated network segments with deliberately misconfigured services. Phishing detection was evaluated against a curated dataset drawn from PhishTank and a corresponding set of benign URLs. Secrets detection evaluation used a repository corpus containing intentionally planted credentials, API keys, and private tokens.

B. Evaluation Metrics

Performance was measured using three standard information retrieval metrics. Precision (TP / (TP + FP)) captures how often flagged items are genuine threats. Recall (TP / (TP + FN)) measures how completely the system surfaces actual threats. The F1-score the harmonic mean of precision and recall provides a single balanced figure. Evaluation was conducted independently across eight attack scenario categories that together span ScanAI's operational range.

C. Performance Results

Table I summarises per-category results. Across all eight categories, ScanAI returned an average precision of 96.6%, recall of 95.8%, and F1-score of 96.1%.

Secrets leakage detection led the field with an F1-score of 98.7%, a result that reflects both the breadth of Titus's 459-rule detection engine and the live credential validation step that eliminates a substantial share of false positives. Phishing detection came in second at 97.9%, benefiting from the fusion of signals across VirusTotal, URLScan, and domain age data — no single source would achieve the same accuracy alone. Lateral movement detection, at 93.2%, was the most challenging category, largely because legitimate and adversarial administrative traffic look similar in a simulated environment; this remains an area for improvement.

TABLE I: Performance Evaluation Across Attack Scenario Categories

Attack Scenario	Precision	Recall	F1-Score
Reconnaissance	96.4%	95.8%	96.1%
Initial Compromise	94.7%	93.9%	94.3%
Payload Execution	97.2%	96.5%	96.8%
Privilege Escalation	95.1%	94.4%	94.7%
Lateral Movement	93.8%	92.7%	93.2%
Data Exfiltration	97.8%	97.1%	97.4%
Phishing Detection	98.3%	97.6%	97.9%
Secrets Leakage	99.1%	98.4%	98.7%
Overall Average	**96.6%**	**95.8%**	**96.1%**

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

D. Comparative Analysis

Table II places ScanAI alongside three widely deployed tools: Burp Suite, OWASP ZAP, and Nessus. The differences are substantial. ScanAI is the only tool in the comparison that offers natural language-driven scan planning, fully automated scanner chaining, cross-scanner attack path identification, and AI-generated reporting. Its 23 scanner modules exceed all three competing tools in coverage, and the 106 configurable YAML profiles provide a degree of operational flexibility that conventional tools simply do not offer. Nessus does provide a composite risk score and partial AI planning capability, but neither extends beyond its own scanner ecosystem.

TABLE II: Comparative Analysis: ScanAI vs. Conventional Tools

Feature	Burp Suite	OWASP ZAP	Nessus	ScanAI
NL Interface	No	No	No	Yes
AI Scan Planning	No	No	Partial	Yes
Auto-Chaining	No	No	No	Yes
Secrets Detection	Plugin	Plugin	Partial	Yes (459 rules)
Cross-Scanner Correl.	No	No	Partial	Yes
Attack Path ID	No	No	No	Yes
Risk Score (0-100)	No	No	Yes	Yes
AI Report Gen.	No	No	No	Yes
Scanner Modules	~10	~8	~15	23
Workflow Profiles	N/A	N/A	N/A	106 YAML

VI. DISCUSSION

The results support the central claim behind ScanAI: that LLM-driven orchestration can automate a substantial portion of the penetration testing lifecycle without meaningful loss of detection quality. An F1-score of 96.1% across eight diverse attack scenarios is not a narrowly optimised result — it holds across categories as different as secrets leakage, phishing detection, and lateral movement, which suggests that the multi-scanner fusion approach is genuinely robust rather than tuned for a particular kind of target.

One architectural choice that has held up well in practice is the clean separation between intelligence and execution. Gemini handles planning and language understanding; the YAML workflow engine handles tool invocation. This means the system can be updated to handle new attack techniques simply by adding or modifying YAML profiles — no model retraining required. Security teams can also extend the framework with their own profiles for custom or proprietary tools, which gives the architecture a longevity that tightly coupled designs often lack.

The Smart Correlation Engine addresses something that individual scanners fundamentally cannot: the ability to turn a pile of disconnected findings into a coherent attack narrative. A missing WAF is a low-severity finding in isolation.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

Combined with an open service port, a matching CVE, and a web application vulnerable to injection, it becomes part of a plausible exploitation chain. No single scanner surfaces that picture; only cross-domain correlation does.

There are genuine limitations worth naming. The current implementation is tied to Kali Linux, which means deployment requires a dedicated security workstation rather than a general-purpose machine. Novel zero-day vulnerabilities without existing signatures will still go undetected — no signature-based system avoids that problem entirely. And reliance on the Google Gemini API introduces a latency floor in the query interpretation step that may be a constraint in time-critical operations.

VII. FUTURE WORK

Several directions look promising for extending ScanAI. Reinforcement learning-based adaptive chain planning is the one we are most interested in exploring: rather than following a fixed scan sequence, the system would adjust dynamically based on intermediate findings, allocating more attention to promising attack surfaces as they emerge. This could substantially improve efficiency in complex environments where the interesting vulnerabilities are rarely where you first look.

Broadening platform support to Docker containers and cloud-native environments — AWS, Azure, GCP — would make the framework usable in a much wider range of operational contexts. A fine-tuned, security-domain LLM as a complement or replacement for the general-purpose Gemini model could sharpen intent interpretation and improve report quality for highly technical scenarios. Integration with threat intelligence platforms like MISP and OpenCTI would allow ScanAI to contextualise findings against known active threat actor techniques and procedures, adding a layer of intelligence that static CVE data cannot provide. Finally, a web-based dashboard would bring the tool within reach of analysts who do not work from the command line.

VIII. CONCLUSION

ScanAI brings together capabilities that have previously only coexisted in a skilled analyst's head: the ability to understand a security objective in plain language, select and sequence the right tools, interpret what they return, and produce guidance that is useful to both technical teams and organisational leadership. By integrating Google Gemini with 23 scanner modules, 106 YAML workflow profiles, nine pre-built attack chains, and a Smart Correlation Engine, it delivers multi-domain security assessments from a single sentence of natural language input.

The experimental results — 96.6% average precision, 95.8% recall, 96.1% F1-score across eight attack scenario categories — confirm that this level of automation does not come at the cost of detection quality. The system consistently outperforms conventional tools on automation depth, scanner coverage, and cross-domain intelligence synthesis.

More broadly, ScanAI points toward what fully autonomous cybersecurity operations could look like as LLMs continue to mature. The gap between the tools available to well-resourced security teams and those within reach of smaller organisations has always been wide; work like this begins to close it. The project is available as an open-source resource, positioned as a foundation for further research into LLM-driven autonomous security systems.

IX. ACKNOWLEDGMENT

The authors thank the faculty and management of Vivekanandha College of Technology for Women, Tiruchengode, for their continued support and encouragement throughout this research.

REFERENCES

1. Google, "Gemini: A family of highly capable multimodal models," Google DeepMind Technical Report, 2023.
2. R. Deraison, "Nessus: An open-source network vulnerability scanner," USENIX Security Symposium, 2000.
3. Greenbone Networks, "OpenVAS: Open Vulnerability Assessment System," [Online]. Available: https://www.openvas.org, 2023.
4. OWASP Foundation, "OWASP ZAP: Zed Attack Proxy," [Online]. Available: https://www.zaproxy.org, 2023.
5. S. M. Ghaffarian and H. R. Shahriari, "Software vulnerability analysis and discovery using machine learning and data mining techniques: A survey," ACM Computing Surveys, vol. 50, no. 4, pp. 1-36, 2017.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802066

6. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
7. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
8. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
9. S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
10. S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
11. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
12. C. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
13. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
14. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
15. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
16. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
17. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
18. Z. Feng et al., "CodeBERT: A pre-trained model for programming and natural languages," in Proc. EMNLP Findings, 2020, pp. 1536-1547.
19. G. Deng et al., "PentestGPT: An LLM-empowered automatic penetration testing tool," in Proc. USENIX Security, 2024.
20. A. Samtani, K. Chinn, C. Larson, and H. Chen, "AZSecure hacker assets portal: Cyber threat intelligence and malware analysis," in Proc. IEEE ISI, 2016, pp. 19-24.
21. S. Pozdniakov et al., "Smart security audit: Reconnaissance and vulnerability assessment using open-source intelligence," in Proc. IEEE MIPRO, 2020, pp. 1128-1133.
22. Z. Wang et al., "A multi-agent framework for automated penetration testing with large language models," arXiv:2401.00151, 2024.