WiFiPenTester: Towards Governed GenAI-Assisted Wireless PenTesting

Haitham S. Al-Sinani[1,3], Chris J. Mitchell[2], Abdulaziz S. Al-Hosni[1], and Sultan T. Al-Harrasi[4]

[1] Diwan of Royal Court, Muscat, Oman. [hsssinani,ashosni]@diwan.gov.om
[2] Royal Holloway, University of London, Egham, UK. C.Mitchell@rhul.ac.uk
[3] German University of Technology in Oman, Muscat, Oman.
Haitham.AlSinani@gutech.edu.om
[4] University of Technology and Applied Sciences, Muscat, Oman.
Sultan.AlHarrasi@utas.edu.om

Abstract. Wireless ethical hacking relies heavily on skilled practitioners manually interpreting reconnaissance results and executing complex, time-sensitive sequences of commands to identify vulnerable targets, capture authentication handshakes, and assess password resilience; a process that is inherently labour-intensive, difficult to scale, and prone to subjective judgement and human error. To help address these limitations, we propose `WiFiPenTester`, an experimental, governed, and reproducible system for GenAI-enabled wireless ethical hacking. The system integrates large language models into the reconnaissance and decision-support phases of wireless security assessment, enabling intelligent target ranking, attack feasibility estimation, and strategy recommendation, while preserving strict human-in-the-loop control and budget-aware execution. We describe the system architecture, governance mechanisms, prompt-engineering methodology, and empirical experiments conducted across multiple wireless environments. The results show that GenAI assistance improves target selection accuracy and overall assessment efficiency, while maintaining auditability and ethical safeguards.

Keywords: AI · GenAI · LLM · Wireless Security · Penetration Testing

1 Introduction

Wireless penetration testing (PenTesting) is essential for assessing the security posture of modern IEEE 802.11 networks, underpinning enterprise connectivity, public infrastructure, and critical services. Despite significant improvements since the era of WEP and early WPA, wireless deployments continue to suffer from weak or reused pre-shared keys, insecure client behaviour, poor access point (AP) configuration, residual support for legacy protocols, and insufficient monitoring of association and deauthentication events. Wireless PenTesting, therefore, remains a critical component of organisational security assurance.

In practice, wireless ethical hacking is heavily dependent on skilled practitioners manually interpreting reconnaissance results, selecting viable targets,

configuring monitor-mode interfaces, orchestrating active deauthentication and handshake capture procedures, and assessing password resilience through offline cracking. These tasks must be performed in dynamic radio-frequency (RF) environments where client presence, channel conditions, and signal quality fluctuate continuously, requiring substantial expertise and situational awareness. As a result, wireless security assessment is labour-intensive, difficult to scale, and prone to subjective judgement and operational error.

To reduce operator burden and improve consistency, existing toolchains, such as `Aircrack-ng`[5] [16] and `Wifite`[6] [13], provide scripted workflows for scanning, handshake capture, and dictionary-based key recovery. While effective, such tools rely almost entirely on static heuristics or human intuition for target selection, attack feasibility estimation, and strategy configuration, and offer limited support for structured reasoning, auditability, or reproducibility of decision processes. As wireless environments become denser and more heterogeneous, these limitations increasingly constrain both research experimentation and professional PenTesting practice.

Recent advances in Generative Artificial Intelligence (GenAI), particularly Large Language Models (LLMs), have demonstrated strong capabilities in contextual reasoning, structured decision support, and adaptive strategy generation across a wide range of domains. In cybersecurity, early studies have explored LLMs for vulnerability analysis, malware classification, exploit generation, and automated PenTesting workflows [1–7, 9, 10, 14, 21, 31, 39]. However, the integration of GenAI into live wireless PenTesting remains largely under-explored, particularly in settings that demand strict safety guarantees, legal compliance, cost control, and human oversight.

To address this gap, we present `WiFiPenTester`, a system for governed and reproducible GenAI-enabled wireless ethical hacking. The system integrates LLM-based reasoning into the reconnaissance and decision-support phases of wireless security assessment, enabling intelligent target ranking, attack feasibility estimation, and strategy recommendation from structured scan metadata, while preserving strict human-in-the-loop (HITL) control over all active and potentially disruptive operations. Unlike prior automation tools, `WiFiPenTester` explicitly enforces budget-aware execution, mandatory operator approval, monitor-mode opt-in verification, structured evidence logging, and prompt persistence to support auditability and experimental reproducibility.

- **RQ1:** To what extent can GenAI assist in selecting viable wireless targets and attack strategies from reconnaissance data under human supervision?
- **RQ2:** How does structured prompt engineering and decision framing influence the accuracy, consistency, and safety of LLM-driven recommendations in wireless PenTesting contexts?
- **RQ3:** What practical limitations arise when deploying LLM-based reasoning within live, time-constrained, and non-deterministic RF environments?

[5] `https://www.aircrack-ng.org/`
[6] `https://www.kali.org/tools/wifite/`

To explore these questions, we designed and implemented `WiFiPenTester` on a Kali Linux testbed equipped with commodity wireless adapters, and tested it across multiple, controlled wireless environments. The system performs governed monitor-mode validation, passive scanning, GenAI-assisted target prioritisation, controlled deauthentication and handshake capture, and dictionary-based password assessment, while recording detailed execution traces and model interactions. Our results demonstrate that GenAI support can substantially improve target prioritisation accuracy and operational efficiency, while also revealing important sensitivities related to prompt structure, incomplete environmental context, and fluctuating wireless conditions.

This paper makes the following contributions: **C1:** a governed GenAI model for wireless PenTesting enforcing bounded autonomy via HITL checkpoints and strict LLM–execution separation; **C2:** demonstrating the feasibility of GenAI-assisted wireless PenTesting under human oversight; **C3:** the design and proof-of-concept (PoC) implementation of `WiFiPenTester`[7], a modular GenAI-enabled wireless network reconnaissance and decision-support system; **C4:** a structured prompt-engineering and output-constraining approach for wireless target ranking and feasibility estimation, including deterministic JSON schemas, prompt persistence, and budget-aware token/cost gating for auditable model interactions; **C5:** an evidence-centric, reproducible wireless assessment experimental workflow with session-scoped archiving of scans, metadata, LLM transcripts, command traces, and validations; **C6:** a critical analysis of design choices, benefits, and limitations, covering governance trade-offs, model reliability, protocol-dependent feasibility, and safe GenAI deployment; and **C7:** an empirical evaluation in authorised, controlled wireless testbeds using commodity hardware and standard tools, showing improved target selection efficiency and consistency, and documenting practical failure modes and operational challenges.

The remainder of this paper is organised as follows. Section 2 reviews related work, while section 3 provides the background. Section 4 describes the operational workflow of `WiFiPenTester`. Section 5 discusses the system's design and governance rationale. Section 6 outlines the prototype and experimental setup. Section 7 analyses the benefits, limitations, and broader implications of the proposed approach. Finally, Section 8 concludes the paper and outlines future work.

2 Related Work

The application of AI to cybersecurity has received significant research attention, spanning intrusion detection, malware analysis, vulnerability discovery, and offensive security, including ethical hacking. Despite notable progress in automated PenTesting and AI-assisted exploitation [1,14,21,31,39], the development of fully autonomous, reliable, and comprehensive PenTesting systems remains an open challenge, particularly in dynamic environments such as wireless networks.

[7] In accordance with responsible disclosure principles, only non-sensitive components will be released as open source (`https://github.com/DrHaitham/WiFiTesterPP`).

Traditional wireless security assessment tools, including `Aircrack-ng`[8] [16], `Reaver`[9] [35], `Kismet`[10] [23], and `Wifite`[11] [13], provide some level of automation for scanning, handshake capture, and key recovery. However, these systems rely primarily on static heuristics, manual target selection, and operator intuition when determining attack feasibility and prioritisation. They offer limited support for structured reasoning, adaptive strategy selection, or traceable decision processes, and do not incorporate modern AI-based analysis techniques.

Recent work has explored the use of LLMs to support offensive security tasks. PentestGPT [14] introduced an LLM-powered PenTesting assistant employing modular reasoning, generation, and parsing components within a HITL workflow. It utilises PenTest Task Trees (PTTs) to preserve context across interactions and mitigate reasoning drift. However, PentestGPT operates primarily as a guided assistant, requiring manual command execution and feedback, thereby limiting the level of automation it can achieve.

Several systems have emerged in parallel with our own work, which investigated LLM-driven PenTesting automation, including multi-agent architectures and autonomous exploit orchestration [21,25]. These efforts demonstrate the feasibility of LLM-guided attack planning in wired network and application-layer contexts but do not explicitly consider the temporal, probabilistic, and client-dependent characteristics of wireless environments, nor the legal and operational risks associated with uncontrolled active wireless attacks.

Our own prior research examined the role of GenAI in ethical hacking across multiple phases of the PenTesting lifecycle. In [9], we proposed a conceptual system for integrating GenAI into PenTesting workflows. Subsequent studies evaluated GenAI-driven assistance in Windows environments [2] and Linux platforms [10], as well as its application to manual exploitation and privilege escalation [3], alongside a broader survey of LLM-powered PenTesting systems [8]. These works demonstrated that LLMs can enhance analyst productivity, support reasoning under uncertainty, and assist in multi-stage attack planning when properly constrained.

Building on this foundation, `PenTest++` [4, 5] introduced a user-centric GenAI-powered system for automating large portions of the PenTesting workflow while maintaining human oversight. `PenTest2.0` [6, 7] further extended this approach by supporting autonomous, multi-turn privilege escalation driven by LLM reasoning, real-time command execution, structured output parsing, persistent task tracking, and cost-aware governance within a HITL model.

In contrast to prior work, this study focuses on the wireless domain and introduces `WiFiPenTester`, a GenAI-enabled system for target selection and feasibility analysis in wireless networks. Extending our earlier GenAI-assisted PenTesting platforms, the system deliberately restricts LLM use to reconnaissance and decision support, with all active operations subject to explicit human

[8] `https://www.aircrack-ng.org/`
[9] `https://www.kali.org/tools/reaver/`
[10] `https://www.kali.org/tools/kismet/`
[11] `https://www.kali.org/tools/wifite/`

approval to ensure reproducibility, auditability, cost control, and responsible deployment. This governance-first design is motivated by the unique challenges of wireless assessment — including partial observability, time-varying conditions, client mobility, and potential impact on third-party devices — which amplify the risks of overconfident or hallucinated LLM outputs. Unlike `PenTest++` and `PenTest2.0`, which operate in comparatively stable, host-centric *wired* networks, `WiFiPenTester` targets pre-access wireless PenTesting, where feasibility is probabilistic, time-sensitive, and shaped by both protocol configuration and volatile physical-layer conditions, making bounded autonomy and strict separation between reasoning and execution core architectural requirements.

3 Background

3.1 IEEE 802.11 Security Fundamentals

Wireless local area networks (WLANs) based on IEEE 802.11 enable mobility but expose communication to both passive eavesdropping and active interference. Security mechanisms have evolved substantially over the last two decades, progressing from the fundamentally flawed Wired Equivalent Privacy (WEP) [12] protocol to Wi-Fi Protected Access (WPA), WPA2 [24], and more recently WPA3 [34, 38].

WEP relies on the RC4 stream cipher with short initialisation vectors (IVs), enabling practical key recovery through statistical analysis of IV reuse, rendering the protocol obsolete shortly after deployment [12]. WPA introduced TKIP (Temporal Key Integrity Protocol) as an interim mitigation, while WPA2 standardised AES-CCMP for confidentiality and integrity protection [29]. The security of WPA/WPA2-PSK (Pre-Shared Key) networks depends on the entropy of the shared passphrase and on client behaviour, since captured authentication material can support offline password guessing.

In parallel, 802.11 management frames (e.g., for deauthentication and disassociation) were historically unauthenticated, enabling denial-of-service (DoS) and handshake-forcing attacks [11]. Although 802.11w introduced Management Frame Protection (MFP), real-world deployment remains inconsistent, and many environments still permit unauthenticated deauthentication traffic [11].

WPA3 [38] was introduced to address key WPA2 weaknesses, notably offline dictionary attacks after handshake capture, by replacing PSK with Simultaneous *Authentication of Equals* (SAE) — a password-authenticated key exchange (PAKE) designed to prevent offline password guessing without AP interaction [22]. In practice, WPA3 introduces distinct risks: online guessing remains possible without robust rate limiting and anti-clogging, and transitional (WPA2/WPA3 mixed) modes can enable downgrade or client-driven fallback attacks, as demonstrated by the Dragonblood results [34]. These findings show that WPA3 security depends on correct configuration and implementation, not merely SAE adoption. Consequently, ethical wireless assessment should also cover configuration verification, downgrade resilience, MFP and client behaviour checks,

and operational controls [20,30], motivating governance-aware feasibility estimation based on protocol mode, client capabilities, and deployment posture.

3.2 Wireless Penetration Testing Workflow

As reviewed in Section 2, established wireless toolchains such as `aircrack-ng`, `airodump-ng`, `aireplay-ng`, `Reaver`, and `Wifite` automate low-level tasks including packet capture, frame injection, and cryptographic verification, but leave core workflow decisions to the operator. Choices such as target prioritisation, timing and channel selection, and balancing operational impact against objectives remain judgement-intensive, while integrated frameworks like `Wifite` rely mainly on simple, heuristic indicators (e.g., signal strength, encryption type, WPS presence) that degrade in dense, dynamic RF environments. Our preliminary lab-based tests indicate that existing tools, notably `Wifite`, offer little or no explanation for target recommendations, and more generally lack GenAI-assisted reasoning and governance support: they record intermediate decisions inconsistently and lack structured context capture, hindering transparency, reproducibility, and auditability. As wireless environments scale in size and complexity, these limitations increase analyst workload and motivate AI-driven decision support under explicit human control.

3.3 Received Signal Strength Indicator (RSSI)

RSSI is a widely used metric that quantifies the power level of a radio signal as observed by a receiving wireless interface [27]. Although it is not standardised to a single absolute scale across chipset vendors, RSSI is commonly expressed in decibel-milliwatts (dBm) and serves as a practical indicator of link quality, propagation conditions, and physical proximity between a station and an AP.

In wireless security assessment, RSSI plays a central role in estimating the feasibility and reliability of packet capture, client interaction, and handshake acquisition, as weak signal conditions increase frame loss, timing instability, and decoding errors. Consequently, PenTesting tools routinely use RSSI thresholds to prioritise targets, select channels, and adapt capture strategies.

Table 1 summarises typical interpretations of RSSI ranges in IEEE 802.11 environments and their operational implications for wireless attacks and monitoring activities. In `WiFiPenTester`, RSSI is treated as a key feature during structured metadata aggregation and GenAI-assisted target ranking, enabling the system to favour networks that are not only cryptographically weaker but also operationally reachable under realistic RF conditions.

4 WiFiPenTester Operation

`WiFiPenTester` is a governed GenAI-assisted wireless PenTesting system that augments WiFite-style workflows with structured decision support, human oversight, and full execution traceability. It operates as follows (see Fig. 1).

Table 1: Typical interpretation of RSSI values in IEEE 802.11 networks

RSSI Range (dBm)	Signal Quality	Operational Implication
≥ -30	Excellent	Very strong signal; stable communication and reliable packet capture expected.
From -31 to -50	Good	Suitable for sustained monitoring and handshake capture.
From -51 to -70	Weak but usable	Increased frame loss possible; active interaction may be unreliable.
≤ -85	Poor / Unusable	High packet loss; handshake capture and active attacks are unlikely to succeed.

1. **System readiness validation:** The system verifies wireless interface availability, driver compatibility, regulatory domain configuration, and toolchain dependencies before any RF operation is permitted.
2. **Monitor-mode activation:** The selected wireless interface is transitioned into monitor mode only after explicit user confirmation, ensuring that potentially disruptive behaviour is always intentional and authorised.
3. **Passive reconnaissance and client observation:** Beacon frames and probe responses are collected to enumerate nearby APs, encryption schemes, authentication modes, channels, signal strength indicators, and advertised capabilities. Simultaneously, associated stations and traffic rates are also monitored to estimate temporal feasibility of handshake capture and the likelihood of successful active interaction.
4. **Structured metadata aggregation:** All observed network information is normalised into an internal structured representation that captures the essential properties of each discovered wireless network. This includes the Basic Service Set Identifier (BSSID) and the Extended Service Set Identifier (ESSID), the encryption and authentication configuration (e.g., WEB, WPA2, WPA3, or WPS), the operating channel and the RSSI (see Section 3.3), the number of associated client stations, short-term traffic activity levels, and relevant protocol features such as the presence of MFP.
5. **Prompt construction:** The system injects the aggregated metadata into a predefined prompt template that:
   - explicitly assigns the LLM an expert role ("a seasoned wireless penetration tester") to frame its reasoning style and domain assumptions;
   - restricts the LLM to advisory reasoning only (i.e., target ranking and feasibility analysis, without triggering actions);
   - enforces a deterministic, structured JSON output schema with fixed fields for scores, justification, confidence, and recommended targets;
   - defines task-specific scoring criteria and interpretation guidelines to reduce ambiguity and improve consistency of model recommendations in dense wireless environments;
   - provides session context (timestamps, session identifier, tool name, etc.) to ensure traceability and reproducibility across runs; and

- embeds the authorised testing context and scope of engagement, and encodes legal, ethical, and operational constraints.

6. **Budget estimation and human approval:** Token count and estimated API cost are computed and presented to the user together with the full prompt. Only after explicit approval is the prompt submitted to the LLM.
7. **LLM-based target ranking and feasibility estimation:** The LLM returns a structured response containing: ranked candidate targets; feasibility scores; justifications grounded in observed metadata; and qualitative risk indicators. This raw response is persisted verbatim, and a validated JSON parse is generated for internal use.
8. **HITL target selection:** The operator reviews both the model's recommendation and the raw scan data before selecting the actual target. This step enforces bounded autonomy: GenAI advises, but never decides.
9. **Controlled active interaction:** Only after human selection does the system proceed to channel locking, optional deauthentication, and handshake capture, using conventional tools under strict logging and attribution.
10. **Protocol-aware and key-strength assessment:** Captured authentication exchanges are first validated for structural completeness and protocol correctness to ensure that subsequent analysis is meaningful and reproducible. The validation procedure is protocol-aware and is designed to proceed as follows[12].

    - **WEP:** Captured IV-rich traffic traces are inspected to confirm sufficient packet volume and entropy for statistical key recovery attacks. When adequate data is available, automated key recovery may be attempted using standard FMS[13]/Korek-style techniques to evaluate whether the network remains vulnerable to legacy cryptographic weaknesses.
    - **WPA/WPA2-PSK:** Four-way handshakes are verified for completeness and temporal consistency. When a valid handshake is confirmed, dictionary-based assessment may be performed using the user-specified wordlist to evaluate passphrase resilience against offline guessing attacks. The outcome is recorded as either *recovered*, *not recovered*, or *insufficient evidence*, together with timing and attempt statistics.
    - **WPA3-SAE:** As offline key recovery is computationally infeasible by design, `WiFiPenTester` instead evaluates the security posture of the deployment, including: confirmation of SAE usage versus transitional (mixed WPA2/WPA3) modes; detection of downgrade exposure; assessment of MFP (802.11w); and identification of configuration patterns associated with known weaknesses (e.g., Dragonblood-class limitations).

[12] This protocol-aware treatment prevents misleading generalisations across heterogeneous security mechanisms and enables the system to report wireless security posture in a manner that accurately reflects cryptographic design intent, operational constraints, and realistic attacker capabilities.

[13] FMS [17] refers to the Fluhrer–Mantin–Shamir attack, a statistical RC4 key-recovery attack that exploits weak IVs in WEP.

11. **Evidence consolidation:** All operational artefacts are collected, including: raw scan outputs; structured network metadata; LLM prompts and responses; token usage and costs; executed commands & timestamps; handshake status; and assessment outcomes.
12. **GenAI-assisted report generation:** At the conclusion of the assessment, `WiFiPenTester` constructs a final reporting prompt containing all relevant technical facts, observations, and outcomes, and submits it to the LLM to generate a structured PenTesting report in JSON format.
13. **Termination and archival:** The workflow terminates upon completion of reporting, timeout, or user intervention. All artefacts are archived to support reproducibility, auditability, and subsequent research analysis.

5 Design Rationale

5.1 Human-Governed Active Interaction and Operational Ordering

The ordering of steps in Section 4 is itself a design decision. Passive reconnaissance and client observation are conducted first, providing a factual basis for feasibility reasoning and limiting unnecessary transmissions. Active interaction is placed only after target selection and explicit user approval. This ordering ensures that disruptive actions are justified by observed conditions (e.g., the presence of clients, sufficient signal quality, and plausible handshake capture opportunities). It also reflects a practical constraint of wireless assessments: feasibility is time-dependent and context-sensitive, and therefore cannot be reduced to static heuristics. By design, `WiFiPenTester` allows GenAI to assist with prioritisation and feasibility estimation, but it maintains explicit operator approval of any operation that might affect availability.

5.2 Bounded Autonomy as a Hard Requirement

Wireless PenTesting operates in a shared, RF environment where active actions may affect third-party devices and production services. In contrast to post-exploitation tasks performed within an assumed-breach host context (as, e.g., in PenTest2.0 [6, 7]), wireless operations such as deauthentication and channel locking can cause observable disruption and may carry heightened legal and ethical exposure. For this reason, `WiFiPenTester` adopts bounded autonomy as a non-negotiable requirement: GenAI provides recommendations and structured reasoning, but the operator retains sole authority over decisions and all active transmissions. Concretely, the workflow enforces explicit user approval before each LLM invocation and before any active wireless action, ensuring that model output cannot directly trigger potentially disruptive behaviour. This decision is motivated by the need for operational safety, scope compliance, and accountable use of offensive capabilities in real environments.

5.3 Prompt Design as Rules of Engagement

As described in Section 4 (Step 5), LLM behaviour is highly sensitive to prompt structure, context, and constraints [15,26,28,32]. Accordingly, `WiFiPenTester` treats prompt construction as an explicit mechanism for defining the system's *rules of engagement*, rather than a formatting step. The model is assigned the role of a 'seasoned wireless penetration tester' to anchor domain-specific reasoning and protocol-aware assessment, while being strictly constrained to an advisory function limited to target ranking and feasibility analysis. Action generation is prohibited to preserve human control, and structured JSON outputs with fixed scoring criteria are enforced to reduce ambiguity, support comparability, and enable auditing. Session metadata and explicit legal and ethical constraints are embedded in the prompt to ensure traceability and reproducibility, collectively transforming it into a governed interface that bounds model behaviour and supports safe GenAI integration in security-critical workflows.

The scoring criteria in `WiFiPenTester` formalise target selection as a deterministic, weighted decision process in which each detected network is assigned a 0–100 score based on *Vulnerability* (40%), *Accessibility* (30%), and *Value* (30%). *Vulnerability* maps protocol and configuration exposure to baseline scores (e.g., WEP, WPA, WPA2, WPA3) with explicit modifiers for risk factors such as TKIP usage, WPS availability, and generic ESSIDs, while *Accessibility* captures environmental feasibility via observable RF indicators (RSSI and active clients), and *Value* encodes contextual relevance by prioritising higher-impact targets (e.g., corporate or public networks). For each target, the LLM must compute and justify the score, enumerate attacker-relevant strengths and weaknesses, and apply explicit GO/CAUTION/NO-GO logic, ensuring transparent, auditable, and human-governed decision support rather than autonomous action.

5.4 Use of Chain-of-Thought (CoT) Reasoning

Recent work [37, 40] has shown that CoT prompting can significantly improve the quality and consistency of LLM reasoning by encouraging models to decompose complex decisions into intermediate analytical steps. In the context of wireless PenTesting, such decomposition is particularly valuable, as target prioritisation and feasibility estimation depend on multiple interacting factors, including protocol configuration, signal quality, client activity, and environmental stability. `WiFiPenTester` therefore adopts CoT reasoning in a constrained and explicitly governed manner: the system's prompt templates instruct the LLM to reason step-by-step over the structured scan metadata before producing its final recommendation, thereby improving coherence and reducing shallow heuristic decisions, while remaining limited to advisory analysis only and prohibited from generating commands, triggering actions, or altering execution flow.

5.5 Reasoning over Structured Observations, not Free-Form Logs

A central objective of `WiFiPenTester` is to ensure that LLM reasoning is grounded in verifiable observations rather than ambiguous or stylistically vari-

able tool output. Raw scan logs and packet-level artefacts are often noisy, vendor-dependent, and difficult to interpret consistently, both for humans and for LLMs. Accordingly, the system normalises reconnaissance results into a structured internal representation prior to prompt construction. This representation includes only the fields required for decision support (e.g., security mode, channel, RSSI, client activity indicators, and relevant protocol features). The prompt is then constructed by injecting these structured facts into a constrained template. As a result, the LLM is compelled to reason over concrete wireless observations rather than free-form descriptions, improving robustness, reducing hallucination risk, and supporting deterministic downstream parsing of the LLM's JSON response.

5.6 Strict Separation between Reasoning and Execution

`WiFiPenTester` deliberately separates GenAI-driven reasoning from operational execution. All wireless actions are performed by deterministic local tooling (e.g., `airodump-ng`, `aireplay-ng`, `aircrack-ng`), while the LLM is confined to advisory functions such as target ranking, feasibility estimation, qualitative risk assessment, and final report generation. This separation mitigates prompt-injection and instruction-following risks by preventing model output from becoming executable control flow. It also preserves experimental repeatability: the execution path remains attributable to known tools and parameters, while the model's role is clearly scoped to analysis and documentation. In practice, this design is essential for rigorous evaluation, as LLM variability is isolated from radio operations.

5.7 Privacy-Preserving GenAI Integration

As introduced in Section 4 (Step 12), `WiFiPenTester` employs GenAI both for decision support and for assisting in the generation of structured assessment reports. This improves usability and supports consistent documentation; however, it introduces privacy and confidentiality considerations. The reporting design therefore follows a strict data-minimisation policy. Raw packet captures, cryptographic material, and client identifiers are not submitted to the LLM. Most importantly, cracked credentials are never transmitted in plaintext. Instead, the report-generation prompt includes only factual outcomes (e.g., "passphrase recovered" or "dictionary attempt unsuccessful") and may include masked representations where necessary for narrative clarity. This design preserves confidentiality while still enabling the LLM to produce a useful, structured PenTesting report. It also aligns with recognised principles of data minimisation and purpose limitation, and supports deployment in environments with strict governance requirements. In addition, the architecture does not mandate the use of cloud-based LLMs. To further reduce the risk of data exposure, `WiFiPenTester` can be configured to operate with locally deployed LLMs (e.g., `llama.cpp`[14] [18,33],

[14] `https://github.com/ggml-org/llama.cpp`

`Ollama`[15]) executed entirely within the assessment environment. This allows all prompts and generated reports to remain on the assessor's infrastructure, eliminating third-party data transfer and supporting compliance with strict organisational, regulatory, or air-gapped operational constraints. Such deployment is particularly relevant for governmental and critical-infrastructure environments where external data sharing is generally prohibited.

5.8 Cost Awareness as an Explicit System Constraint

Unlike conventional security tooling, LLM usage incurs direct financial cost and risks unbounded prompt growth if unconstrained; accordingly, `WiFiPenTester` treats cost awareness as an explicit operational constraint by computing token counts and estimated monetary cost before each LLM invocation and requiring operator approval. This mechanism prevents unexpected expenditure, discourages prompt bloat, and provides consistent cost telemetry for experimental evaluation, while reinforcing bounded autonomy by keeping the operator accountable for both actions and resource consumption. The design is grounded in prior experience with `PenTest2.0` [6, 7], where the absence of explicit cost control led to prompt inflation and rapid exhaustion of API credits, motivating the adoption of manual cost-approval checkpoints in `WiFiPenTester`.

5.9 LLM Safety Mechanisms

Some LLMs incorporate built-in safety and policy enforcement mechanisms that may refuse or partially redact responses to PenTesting-related prompts; while not observed in our PoC experiments, this remains a practical consideration for GenAI-assisted security research. In practice, such limitations can be mitigated through model substitution, as LLMs differ in calibration and policy interpretation, and an abstracted LLM interface — such as that in `WiFiPenTester` — allows assessors to switch models without modifying the surrounding workflow. Moreover, prior work has shown that LLM safety enforcement is probabilistic and model-dependent rather than formally guaranteed, with documented bypass classes including prompt re-framing, multi-turn context manipulation, role conditioning, and indirect instruction encoding [19, 36, 41]. From a design perspective, these findings show that safety controls cannot replace explicit governance and human oversight, and that model replaceability is a key robustness feature under restrictive or unpredictable provider behaviour.

6 Prototype Implementation

6.1 Experimental Setup

`WiFiPenTester` is implemented in Python 3, selected for its rapid prototyping capabilities, mature ecosystem of networking and wireless-security libraries, and

[15] `https://ollama.com/`

seamless integration with modern LLM APIs. All experiments were conducted using VirtualBox 7 on a physical host system running Windows 11 (Lenovo laptop, Intel Core Ultra 7 CPU, 32 GB RAM). The experimental environment consisted of a Kali Linux virtual machine (VM) hosting `WiFiPenTester` and a collection of external APs deployed for controlled testing.

VirtualBox exposes wireless interfaces as Ethernet devices without access to raw wireless frames, and the host laptop's built-in adapter does not support monitor mode or packet injection; therefore, an external USB Wi-Fi adapter was required. Specifically, a MediaTek-based USB 802.11n adapter, supporting both monitor mode and packet injection, was attached to the Kali VM via USB passthrough[16]. This configuration enabled full passive capture and active interaction, including deauthentication and handshake acquisition, as required by `WiFiPenTester`. Successful operation in monitor mode was verified at runtime (e.g., `wlan0` operating in *Mode: Monitor*), ensuring that low-level wireless frame capture and injection were supported throughout the experiments.

For GenAI integration, the prototype uses OpenAI's `gpt-4o-mini` via the official API, selected for its favourable cost–performance trade-off and stable structured-output behaviour; although the system supports alternative models (e.g., `o3`, `gpt-4.1`, `gpt-5`), these were not used in the reported experiments due to higher token cost and variability. All LLM interactions are handled by a dedicated connector module that enforces prompt validation, cost estimation, and explicit user approval prior to submission. The same modular connector abstracts provider-specific APIs and is designed to facilitate future integration with other alternative models (e.g., Claude (`https://www.anthropic.com/claude`), Gemini (`https://deepmind.google/technologies/gemini/`), etc.) or self-hosted open-weight models (e.g., LLaMA (`https://ai.meta.com/llama/`) or Ollama (`https://ollama.com/`)), supporting portability and reproducible experimentation across deployment configurations.

6.2 PoC Implementation

The prototype implements a governed GenAI-assisted wireless PenTesting system augmenting WiFite-style workflows with structured decision support, human oversight, and execution traceability, implemented as a modular, command-line (CLI) application executed on a Kali Linux VM and operates as follows.

1. **Initialisation:** The system is launched via a single CLI entry point (`cli.py`) with all execution parameters supplied explicitly by the operator, including the wireless interface identifier, scan duration, wordlist path, timeout, GenAI mode, and automation flags. Internally, argument parsing constructs a session configuration object that is propagated across all modules. A readiness phase validates kernel driver availability, USB device passthrough status, regulatory domain configuration, and the presence of required user-space tooling (e.g., `iw`, `airodump-ng`, `aireplay-ng`, `aircrack-ng`). The system also

[16] MediaTek MT7601U chipset (USB vendor ID: 0x148f, product ID: 0x7601).

initialises a structured directory hierarchy for evidence storage and creates a timestamped session identifier to tag all subsequent artefacts. Lightweight checks equivalent to querying interface capabilities (`iw dev`), tool availability (`which airodump-ng`), and USB enumeration are performed before execution continues.

2. **Monitor-mode:** After readiness confirmation, the system requests explicit user consent before enabling monitor mode on the selected interface. This transition is mandatory and intentionally enforced to guarantee deliberate engagement in RF operations. The interface state is programmatically requeried after activation to confirm IEEE 802.11 monitor support, operating channel, transmit power, and MAC address. Network management services that could interfere with raw frame capture (e.g., `NetworkManager`, `wpa_supplicant`) are temporarily suspended. At the command level, this phase corresponds to actions conceptually equivalent to invoking monitor-mode utilities (e.g., `airmon-ng start`, or `ip link set` + `iw dev set type monitor`) followed by state verification via `iwconfig`.
3. **Passive reconnaissance and client observation:** The system then initiates a bounded passive scanning window, during which it listens for beacon frames and probe responses across channels to enumerate nearby APs. No frames are transmitted at this stage. For each discovered BSS (Basic Service Set), the system extracts ESSID/BSSID pairs, channel number, encryption & authentication suites, RSSI, and observed client counts. Traffic rate and frame counters are monitored to approximate short-term activity levels and infer the probability of observing authentication handshakes in later phases. Discovered networks are tabulated alongside their respective encryption protocols, channel assignments, signal power levels, and associated stations. Internally, this corresponds to spawning a passive capture process similar to `airodump-ng` with channel hopping enabled, while continuously parsing its structured output stream.
4. **Structured metadata aggregation:** All raw observations collected during passive scanning are normalised into an internal schema capturing identifiers, protocol properties, signal metrics, and client activity indicators. The resulting dataset is serialised into structured JSON records and persisted to disk as part of the session evidence bundle. Raw capture files (PCAP/CAP) are retained locally for forensic inspection and reproducibility.
5. **Prompt construction:** The aggregated wireless metadata is serialised and injected into a predefined system prompt template that formalises the conditions under which GenAI reasoning is permitted. The template explicitly assigns the LLM the role of a "seasoned wireless penetration tester" in order to frame its reasoning style and domain assumptions, restricts the model to advisory analysis only (target ranking and feasibility assessment), and prohibits the generation of operational commands or autonomous actions. A deterministic JSON output schema is enforced, specifying fixed fields for candidate scores, justification, confidence estimates, and recommended targets, together with task-specific scoring criteria and interpretation guidelines to reduce ambiguity and improve consistency in dense wireless environments.

To ensure traceability and experimental reproducibility, the prompt further embeds session context, including timestamps, session identifiers, tool name, and assessment stage. The authorised testing context and scope of engagement are also encoded directly within the prompt, together with legal, ethical, and operational constraints governing permissible behaviour.

6. **Cost estimation and human approval:** Prior to submission, the system computes the exact token count of the composed prompt and estimates the corresponding monetary cost for the selected model. These values are presented to the operator via an interactive "LLM Budget Gate", which pauses execution until explicit approval is provided (e.g., by responding `y`). Fig. 2 illustrates the cumulative token usage and cost table displayed at this stage.
7. **LLM-based target ranking:** Upon approval, the prompt is submitted to the GenAI backend. The model returns a single structured JSON object containing ranked candidate networks, confidence scores, inferred strengths and weaknesses, and a recommended strategy. The raw response is archived verbatim and then validated against a strict schema before being parsed into internal data structures. Figs. 3 and 4 illustrate the returned JSON object, including ranked ESSIDs, BSSIDs, scores, and reasoning fields such as signal strength, encryption type, client count, and estimated handshake feasibility.
8. **HITL target selection:** The ranked candidate list is presented to the operator alongside the original passive scan table. The operator manually selects the target network by index. GenAI output is advisory only and never triggers wireless transmission or channel manipulation directly. This interaction is visible in Fig. 4, where the user selects the recommended candidate via numeric input. Internally, this simply updates the session context with the chosen BSSID and channel parameters.
9. **Handshake capture:** Only after explicit target selection does the system transition to controlled active operations. The wireless interface is locked to the target channel, a focused capture process is launched for the selected BSSID, and limited authentication-stimulating traffic may be generated to provoke protocol handshakes. All executed actions, timestamps, interface state changes, and process identifiers are logged. At the tooling level, this corresponds to invoking focused capture utilities (e.g., `airodump-ng`[17]) and controlled injection utilities (e.g., `aireplay-ng`[18]) while monitoring output streams for handshake indicators.
10. **Handshake validation and assessment:** Captured authentication exchanges are analysed to verify structural correctness (e.g., message sequence completeness, replay counters, cipher suite consistency). Rather than recording a binary success flag, the system stores detailed validation metadata describing which protocol elements were observed and whether they satisfy formal handshake requirements. This step operates on the capture artefacts created in the previous phase and produces structured validation records linked to the session identifier.

[17] Example invocation used in our experiments: `airodump-ng --bssid AA:BB:CC:DD:EE:FF --channel 6 --write target-01 wlan0`

[18] Example: `aireplay-ng --deauth 10 -a AA:BB:CC:DD:EE:FF wlan0`

When a target network is identified as operating under WPA or WPA2 using PSKs, the system explicitly verifies whether a complete and temporally consistent four-way handshake has been captured. In practice, this requires observing at least message 1 and message 2, and preferably all four messages, to ensure that the pairwise transient key (PTK) derivation can be validated offline. The capture file is inspected to confirm the presence of the `EAPOL` (Extensible Authentication Protocol over LAN) frames corresponding to the handshake sequence and to extract key parameters such as the AP-generated ANonce (Authenticator Nonce), the client-generated SNonce (Supplicant Nonce), MAC addresses, replay counters, and negotiated cipher suite. This validation step prevents false positives arising from partial captures, duplicated frames, or unrelated authentication attempts. If a valid handshake is confirmed, the system proceeds to assess passphrase resilience by invoking deterministic offline verification using standard tooling. The captured trace (e.g., `target-01.cap`) is supplied to a dictionary-based verification process using widely adopted tools such as `aircrack-ng`[19], `John the Ripper (JtR)`, or `hashcat`. This tests candidate passphrases against the captured handshake without any further interaction with the target network. The outcome of this process (e.g., no match found, weak passphrase recovered, or verification inconclusive) is recorded as evidence together with the handshake metadata, timing information, and cryptographic parameters.

11. **Evidence consolidation:** All artefacts generated during the session are consolidated into a single evidence tree, including passive scan summaries, structured metadata JSON files, prompt and response transcripts, token usage logs, executed command traces, timestamps, and handshake validation reports. The directory structure preserves a strict one-session-per-folder layout to support later auditing and controlled replication.
12. **Report generation:** From the consolidated evidence, a second prompt is constructed to generate a structured PenTesting report in JSON format. Sensitive fields (e.g., recovered secrets, raw packet payloads, client identifiers) are excluded or masked prior to submission. As in earlier stages, cost estimation and user approval are enforced before invoking the model.
13. **Termination and archival:** `WiFiPenTester` terminates either after successful completion, timeout expiration, or user interruption. The system restores network services if previously suspended and leaves all artefacts stored locally to support reproducibility, auditing, and later experimental analysis.

6.3 Testing Results

We conducted controlled lab experiments to evaluate the PoC under realistic, bounded conditions. `WiFiPentester` ran on Kali Linux with an external USB adapter supporting monitor mode and packet injection, requesting user approval before monitor-mode activation. The PoC then performed passive discovery of

[19] The command: `aircrack-ng -w rockyou.txt target-01.cap` performs a dictionary check against a WPA2 handshake using the `rockyou.txt` wordlist.

nearby wireless networks, detecting multiple candidates per run (e.g., 10 networks in a test environment configured for evaluation). Prior to each LLM invocation, the operator remained in the loop; `WiFiPentester` displayed a structured table summarising the prompt content together with estimated token usage and cost, and prompt submission proceeded only after explicit user approval (see Fig. 2). The LLM interpreted the provided context and instructions and produced a ranked assessment of the detected networks, encouraged by the scoring criteria and other instructions in the prompt (Section 5.3), ordering them from most to least susceptible. The LLM also provided brief justifications for its ranking and recommended attack strategy, reinforcing the ranking quality. In the absence of WEP-protected wireless networks, higher confidence scores were assigned to WPA/WPA2 networks with active clients and stronger signal strength (better RSSI; see Section 3.3), while networks with no associated clients, poor RSSI, or stronger protection (e.g., WPA3) were given lower rankings (see Fig. 3). `WiFiPentester` then parsed the LLM output and presented the results to the operator in a concise ranked list, with the most susceptible network shown first (see Fig. 4). Upon operator approval, `WiFiPentester` initiated deauthentication attempts against the selected target to stimulate authentication traffic, successfully captured a WPA2 handshake in the tested scenario, and subsequently launched an offline dictionary attack using the `rockyou.txt` wordlist, recovering the passphrase. Finally, the experiments confirmed that `WiFiPentester` correctly deprioritised practically unsuitable targets, while remaining sensitive to dynamic wireless conditions, e.g. fluctuating RSSI, transient APs, and changing client association states, which can influence capture reliability across runs.

7 Discussion and Implications

7.1 Benefits and Limitations

Benefits: `WiFiPenTester` demonstrates that GenAI can be integrated into wireless PenTesting workflows in a controlled, auditable, and practically useful manner by combining structured RF reconnaissance with model-assisted reasoning and strict human oversight, thereby extending traditional toolchains beyond static heuristics while preserving operational safety. First, the system enables GenAI-assisted target prioritisation under explicit human control: the integrated LLM analyses structured scan metadata and ranks candidate networks based on protocol configuration, signal quality, and observed activity, reducing operator cognitive burden in dense wireless environments while ensuring that all recommendations remain strictly advisory and require approval before any active operation. In addition, strict HITL governance is enforced throughout the workflow, with all disruptive actions — including monitor-mode activation, deauthentication, handshake capture, and report generation — gated by explicit operator confirmation, reducing legal and ethical risk and preventing uncontrolled RF interference. Moreover, GenAI usage is made budget-aware and fully auditable by computing token usage and estimated cost prior to each LLM interaction and

archiving all prompts and responses verbatim, supporting reproducibility, post-hoc analysis, and systematic experimentation. To further support experimental rigour and accountability, the system employs structured evidence management, consolidating passive scans, normalised metadata, LLM transcripts, command traces, and validation records into a session-scoped evidence tree that facilitates auditing and comparative evaluation. Furthermore, the prototype adopts a modular, extensible architecture with clearly separated components for wireless interaction, metadata processing, LLM communication, and reporting, enabling straightforward extension to new protocols, hardware, models, or outputs. Finally, operational realism is demonstrated through execution on commodity hardware using standard toolchains such as `aircrack-ng` and low-cost external USB adapters, confirming feasibility under practical deployment constraints rather than simulated or synthetic conditions.

Limitations: There remain limits to `WiFiPenTester` capabilities. First, the system remains dependent on the quality and completeness of passive reconnaissance data. Wireless environments are inherently volatile: client mobility, fluctuating signal strength, hidden SSIDs, and transient associations can lead to incomplete or misleading snapshots. As a result, GenAI recommendations may be based on partial context, occasionally producing overconfident or suboptimal rankings. Second, although strict human approval is enforced, active wireless operations such as deauthentication and handshake capture are intrinsically disruptive and may violate organisational policies or regulatory constraints if misused. The system mitigates this risk through governance controls, but it cannot eliminate the legal responsibility of the operator. Third, the reliance on online LLM APIs introduces privacy and data-protection considerations. While raw packet contents and credentials are excluded from prompts, structured metadata may still reveal sensitive information about network topology or security posture. Deployment in production environments therefore requires careful compliance with institutional and legal data-handling policies. However, as discussed in Section 5.7, this limitation can be mitigated through strict prompt-level data minimisation and the use of locally deployed LLMs to eliminate external data transfer. Fourth, the prototype has been evaluated primarily on WPA/WPA2 networks using dictionary-based verification. Full WPA3-SAE support remains very limited in the current PoC and is left for future work. Finally, the system inherits inherent limitations of LLMs, including sensitivity to prompt phrasing, occasional hallucination, and inconsistent reasoning under distributional shift. Although structured prompting and schema validation reduce these risks, they do not eliminate them, reinforcing the necessity of continued HITL supervision.

7.2 Answers to the Research Questions

We now address the three research questions (RQ) posed in Section 1.

RQ1: Our results indicate that GenAI can provide meaningful and operationally useful support in prioritising wireless targets when supplied with structured scan metadata. The LLM consistently identified networks with stronger

signal quality, active clients, and weaker authentication configurations as higher-feasibility candidates, often aligning with expert judgement. However, this assistance remains probabilistic rather than deterministic, and human validation is essential to account for environmental context that may not be captured in scan data (e.g., physical location, user intent, or legal scope). Thus, GenAI functions effectively as a decision-support layer rather than an autonomous attacker.

RQ2: Structured prompts with explicit output schemas, constrained task definitions, and protocol-aware framing substantially improved both the consistency and safety of model responses. In particular, requiring machine-readable JSON output and embedding operational boundaries reduced hallucinated actions and prevented the model from proposing unauthorised transmissions. Nevertheless, prompt sensitivity remained observable: small changes in metadata ordering or phrasing occasionally altered ranking priorities, highlighting the importance of systematic prompt design and validation in wireless contexts where incomplete information is common.

RQ3: Several limitations were observed. Temporal variability in client activity meant that networks deemed feasible during reconnaissance occasionally failed to yield handshakes during the active phase. Conversely, low-activity networks sometimes became viable unexpectedly. The LLM, operating on static snapshots, could not adapt to such dynamics without additional feedback loops. Furthermore, radio interference, channel congestion, and hardware driver behaviour introduced uncertainties that no purely symbolic reasoning system can fully anticipate. These findings suggest that GenAI integration is most effective when combined with continuous sensing and conservative operational thresholds rather than treated as a standalone decision engine.

Overall, `WiFiPenTester` demonstrates that GenAI can augment wireless PenTesting in a principled and practically useful manner when embedded within a rigorously governed architecture. The system highlights both the promise of LLM-assisted reasoning for complex security workflows and the necessity of strict human oversight, protocol awareness, and evidence-driven validation in safety-critical cyber–physical environments.

8 Conclusions and Further Research

In this paper, we presented `WiFiPenTester`, a system for governed and reproducible GenAI-assisted wireless PenTesting. The system integrates large language model reasoning into the reconnaissance and decision-support phases of wireless security assessment, enabling intelligent target prioritisation, feasibility estimation, and strategy recommendation from structured scan metadata, while preserving strict human-in-the-loop control over all active and potentially disruptive operations. We designed and implemented a proof-of-concept prototype operating on commodity hardware and standard wireless toolchains, and evaluated its behaviour across realistic wireless environments.

Our results demonstrate that GenAI can reduce operator cognitive load during dense wireless reconnaissance, improve consistency in target selection, and

provide structured, auditable reasoning to support tactical decision making. At the same time, the system enforces budget awareness, explicit user consent, protocol-aware validation, and comprehensive evidence logging, addressing key safety, legal, and reproducibility challenges inherent to wireless ethical hacking. Unlike prior automation tools, `WiFiPenTester` formalises decision support as a governed process rather than an opaque heuristic. Nevertheless, our findings also highlight important limitations. Model recommendations remain sensitive to prompt design and incomplete environmental context, wireless conditions introduce unavoidable non-determinism, and current support for WPA3-SAE remains limited. Such constraints reinforce the necessity of human oversight and careful governance when deploying GenAI within live radio-frequency environments.

Overall, `WiFiPenTester` represents a clear step forward towards safe, transparent, and operationally realistic GenAI-enabled wireless PenTesting. By embedding AI intelligence into the planning and evaluation stages — rather than execution itself — the system bridges the gap between manual expertise and automated assistance without compromising ethical or technical control.

Future research will focus on advancing WiFiPenTester by integrating full protocol-aware support for WPA3-SAE and enterprise-scale features like 802.1X/EAP analysis and rogue AP detection. To improve robustness, we will explore adaptive, offline and online model learning mechanisms to better handle fluctuating wireless conditions. Additionally, we plan to conduct quantitative and qualitative studies on human factors — measuring operator trust, workload reduction, and decision accuracy — while benchmarking against existing automation tools. To support deployment in sensitive environments, ongoing development will investigate offline LLM integration, cryptographic prompt integrity, and formal governance models. Collectively, these enhancements aim to transition WiFiPenTester from a proof-of-concept into a principled, secure foundation for GenAI-assisted wireless security research.

References

1. Abu-Dabaseh, F., Alshammari, E.: Automated penetration testing: An overview. In: Proceedings of the 4th international conference on natural language computing, Copenhagen, Denmark. pp. 121–129 (2018), https://airccj.org/CSCP/vol8/csit88610.pdf
2. Al-Sinani, H., Mitchell, C.: Unleashing AI in ethical hacking: A preliminary experimental study. Technical report, Royal Holloway, University of London (2024), https://pure.royalholloway.ac.uk/files/58692091/TechReport_UnleashingAIinEthicalHacking.pdf
3. Al-Sinani, H.S., Mitchell, C.J.: AI-augmented ethical hacking: A practical examination of manual exploitation and privilege escalation in Linux environments. CoRR **abs/2411.17539** (Nov 2024). https://doi.org/10.48550/ARXIV.2411.17539, https://doi.org/10.48550/arXiv.2411.17539
4. Al-Sinani, H.S., Mitchell, C.J.: Introducing PenTest++: an AI-augmented, automated, ethical hacking system. In: Proceedings of the 2nd International Workshop on Cybersecurity: Blockchain and Artificial Intelligence Applications (CyBAI '25), part of the 8th Congress on Information Science and Technology

(CiSt), 4–10 October 2025. pp. 565–570. IEEE, Marrakech, Morocco (Oct 2025), https://ieeexplore.ieee.org/document/11224070
5. Al-Sinani, H.S., Mitchell, C.J.: PenTest++: Elevating ethical hacking with AI and automation. CoRR **abs/2502.09484** (Feb 2025). https://doi.org/10.48550/ARXIV.2502.09484
6. Al-Sinani, H.S., Mitchell, C.J.: PenTest2.0: Towards autonomous privilege escalation using GenAI. CoRR **abs/2507.06742** (jul 2025). https://doi.org/10.48550/ARXIV.2507.06742
7. Al-Sinani, H.S., Mitchell, C.J., Al-Hosni, A.S.: PenTest2.0: Advancing ethical hacking with GenAI-driven privilege escalation. In: Proceedings of the 40th International Conference on Advanced Information Networking and Applications (AINA '26), Wellington, New Zealand, April 8–10, 2026. Lecture Notes in Data Engineering and Communication Technologies (LNDECT), Springer (Apr 2026)
8. Al-Sinani, H.S., Mitchell, C.J., Al-Hosni, A.S.: The rise and rise of LLM-powered PenTesting systems: The State of the Art. In: Proceedings of the International Conference on Cybersecurity, Situational Awareness and Social Media (Cyber Science 2026), London, UK. Springer Proceedings in Complexity (Jun 2026), a longer version is available on ResearchGate: https://doi.org/10.13140/RG.2.2.29192.38408
9. Al-Sinani, H.S., Mitchell, C.J., Sahli, N., Al-Siyabi, M.: Unleashing AI in ethical hacking. In: Martinelli, F., Rios, R. (eds.) Proceedings of Security and Trust Management — 20th International Workshop, STM '24, Bydgoszcz, Poland, September 19–20, 2024. LNCS: Lecture Notes in Computer Science, vol. 15235, pp. 140–151. Springer (2024). https://doi.org/10.1007/978-3-031-76371-7_10
10. Al-Sinani, H.S., Sahli, N., Mitchell, C.J., Al-Siyabi, M.: Advancing ethical hacking with AI: A Linux-based experimental study. In: Costa, G., Montanari, R., Carminati, M., Sciarretta, G. (eds.) Proceedings of the Joint National Conference on Cybersecurity (ITASEC & SERICS '25), February 03–08, 2025, Bologna, Italy. vol. 3962. CEUR-WS (Feb 2025). https://doi.org/10.1007/978-3-031-76371-7_10, https://ceur-ws.org/Vol-3962/paper7.pdf
11. Bellardo, J., Savage, S.: 802.11 Denial-of-Service attacks: Real vulnerabilities and practical solutions. In: Proceedings of the 12th USENIX Security Symposium, Washington, D.C., USA, August 4–8, 2003. pp. 15–28. USENIX Association (2003), https://www.usenix.org/conference/12th-usenix-security-symposium/80211-denial-service-attacks-real-vulnerabilities-and
12. Bittau, A., Handley, M., Lackey, J.: The final nail in WEP's coffin. In: IEEE Symposium on Security and Privacy (S&P '06), 21–24 May 2006, Berkeley, Oakland, CA, USA. pp. 15 pp.–400. IEEE (2006). https://doi.org/10.1109/SP.2006.40, https://ieeexplore.ieee.org/abstract/document/1624028
13. Blumer, D., Kimocoder: Wifite2. Open-source software (2020), https://github.com/derv82/wifite2, accessed: 2026-01-16
14. Deng, G., Liu, Y., Mayoral-Vilches, V., Liu, P., Li, Y., Xu, Y., Zhang, T., Liu, Y., Pinzger, M., Rass, S.: PentestGPT: Evaluating and harnessing Large Language Models for automated penetration testing. In: Proceedings of the 33rd USENIX Security Symposium (USENIX Security '24). pp. 847–864. USENIX Association, Philadelphia, PA, USA (Aug 2024), https://www.usenix.org/conference/usenixsecurity24/presentation/deng
15. Denny, P., Kumar, V., Giacaman, N.: Conversing with copilot: Exploring prompt engineering for solving CS1 problems using natural language. In: Proceedings of the

54th ACM Technical Symposium on Computer Science Education. pp. 1136–1142 (2023). https://doi.org/10.1145/3545945.3569755
16. Devine, C., d'Otreppe de Bouvette, T.: Aircrack-ng. Open-source software (2022), https://www.aircrack-ng.org, stable version 1.7. GitHub repository: https://github.com/aircrack-ng/aircrack-ng. Accessed: 2026-01-16
17. Fluhrer, S., Mantin, I., Shamir, A.: Weaknesses in the key scheduling algorithm of RC4. In: Vaudenay, S., Youssef, A. (eds.) Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography (SAC 2001). Lecture Notes in Computer Science (LNCS), vol. 2259, pp. 1–24. Springer, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45537-X_1, https://rdcu.be/e1tQm
18. Gerganov, G.: llama.cpp: Inference of llama model in pure c/c++. https://github.com/ggerganov/llama.cpp (2023), accessed July 2025
19. Hackett, W., Birch, L., Trawicki, S., Suri, N., Garraghan, P.: Bypassing LLM guardrails: An empirical analysis of evasion attacks against prompt injection and jailbreak detection systems. In: Derczynski, L., Novikova, J., Chen, M. (eds.) Proceedings of the First Workshop on LLM Security (LLMSEC), Vienna, Austria, Aug 1, 2025. pp. 101–114. Association for Computational Linguistics (ACL) (Aug 2025), https://aclanthology.org/2025.llmsec-1.8/
20. Halbouni, A., Ong, L.Y., Leow, M.C.: Wireless security protocols WPA3: A systematic literature review. IEEE Access **11**, 112438–112450 (2023). https://doi.org/10.1109/ACCESS.2023.3322931
21. Happe, A., Cito, J.: Getting pwn'd by AI: Penetration testing with Large Language Models. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23). pp. 2082–2086. ESEC/FSE, ACM, New York, NY, USA (2023). https://doi.org/10.1145/3611643.3613083
22. Harkins, D.: Dragonfly key exchange. RFC 7664 (Nov 2015), https://www.rfc-editor.org/rfc/rfc7664, this RFC defines the PAKE protocol underlying SAE in WPA3
23. Kershaw, M.: Kismet. Open-source software, https://www.kismetwireless.net, source code: https://github.com/kismetwireless/kismet. Accessed: 2026-01-16
24. Lashkari, A.H., Danesh, M.M.S., Samadi, B.: A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). In: IEEE International Conference on Computer Science and Information Technology (ICCSIT '09). pp. 48–52. IEEE (2009). https://doi.org/10.1109/ICCSIT.2009.5234856
25. Lazarov, W., Seda, P., Martinasek, Z., Kummel, R.: Penterep: Comprehensive penetration testing with adaptable interactive checklists. Computers & Security **154**, 104399 (2025). https://doi.org/https://doi.org/10.1016/j.cose.2025.104399
26. Liu, P., Yuan, W., et al.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9) (Jan 2023). https://doi.org/10.1145/3560815
27. Mareco, D.: The MSP guide to Wi-Fi signal strength, design, and delivery. Technical blog article (Oct 2025), https://techgrid.com/blog/wifi-signal-strength, accessed: 2026-01-19
28. Ouyang, L., Wu, J., et al.: Training language models to follow instructions with human feedback. In: Koyejo, S., Mohamed, S., et al. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 27730–27744. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

29. Paterson, K.G., Poettering, B., Schuldt, J.C.N.: Plaintext recovery attacks against WPA/TKIP. In: Cid, C., Rechberger, C. (eds.) 21st International Workshop, Fast Software Encryption (FSE '14), London, UK, March 3–5, 2014. pp. 325–349. Springer, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46706-0_17
30. Reddy, B.I., Srikanth, V.: Review on wireless security protocols (WEP, WPA, WPA2 & WPA3). International Journal of Scientific Research in Computer Science, Engineering and Information Technology **5**(4), 28–35 (2019)
31. Stefinko, Y., Piskozub, A., Banakh, R.: Manual and automated penetration testing. benefits and drawbacks. Modern tendency. In: 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET '16). pp. 488–491. IEEE (2016). https://doi.org/10.1109/TCSET.2016.7452095
32. Strobelt, H., Webson, A., Sanh, V., et al.: Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. IEEE Transactions on Visualization and Computer Graphics **29**(1), 1146–1156 (2022). https://doi.org/10.1109/TVCG.2022.3141234
33. Touvron, H., Lavril, T., et al.: LLaMA: Open and efficient foundation language models. CoRR **abs/2302.13971** (Feb 2023). https://doi.org/10.48550/arXiv.2302.13971
34. Vanhoef, M., Ronen, E.: Dragonblood: Analyzing the Dragonfly handshake of WPA3 and EAP-pwd. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P '20), 8–21 May 2020, San Francisco, CA, USA. pp. 517–533. IEEE (May 2020). https://doi.org/10.1109/SP40000.2020.00031
35. Viehböck, S.: Brute forcing Wi-Fi protected setup (2011), https://sec-consult.com/fileadmin/user_upload/sec-consult/Dynamisch/Blogartikel/Old_Blogposts/sec-consult-kcodes-netsub-viehboeck.pdf, SEC Consult White Paper
36. Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: How does LLM safety training fail? In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems (NeurIPS '23), Sydney, Australia, December 10–16, 2023. vol. 36, pp. 80079–80110. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/fd6613131889a4b656206c50a8bd7790-Paper-Conference.pdf
37. Wei, J., Wang, X., et al.: Chain-of-thought prompting elicits reasoning in Large Language Models. In: Koyejo, S., et al. (eds.) Advances in Neural Information Processing Systems (NeurIPS 2022). vol. 35, pp. 24824–24837. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
38. Wi-Fi Alliance: WPA3 specification, version 3.5. Technical specification (2025), https://www.wi-fi.org/system/files/WPA3%20Specification%20v3.5.pdf, accessed: 2026-01-18
39. Xiong, P., Peyton, L.: A model-driven penetration test framework for web applications. In: 8th International Conference on PST: Privacy, Security and Trust. pp. 173–180. IEEE (2010). https://doi.org/10.1109/PST.2010.5593250
40. Yao, S., et. al: Tree of thoughts: Deliberate problem solving with Large Language Models. In: Oh, A., et. al (eds.) Advances in Neural Information Processing Systems (NeurIPS 2023). vol. 36, pp. 11809–11822. Curran Associates, Inc. (2023)
41. Zou, A., et. al: Universal and transferable adversarial attacks on aligned language models. arXiv (2023). https://doi.org/10.48550/arXiv.2307.15043

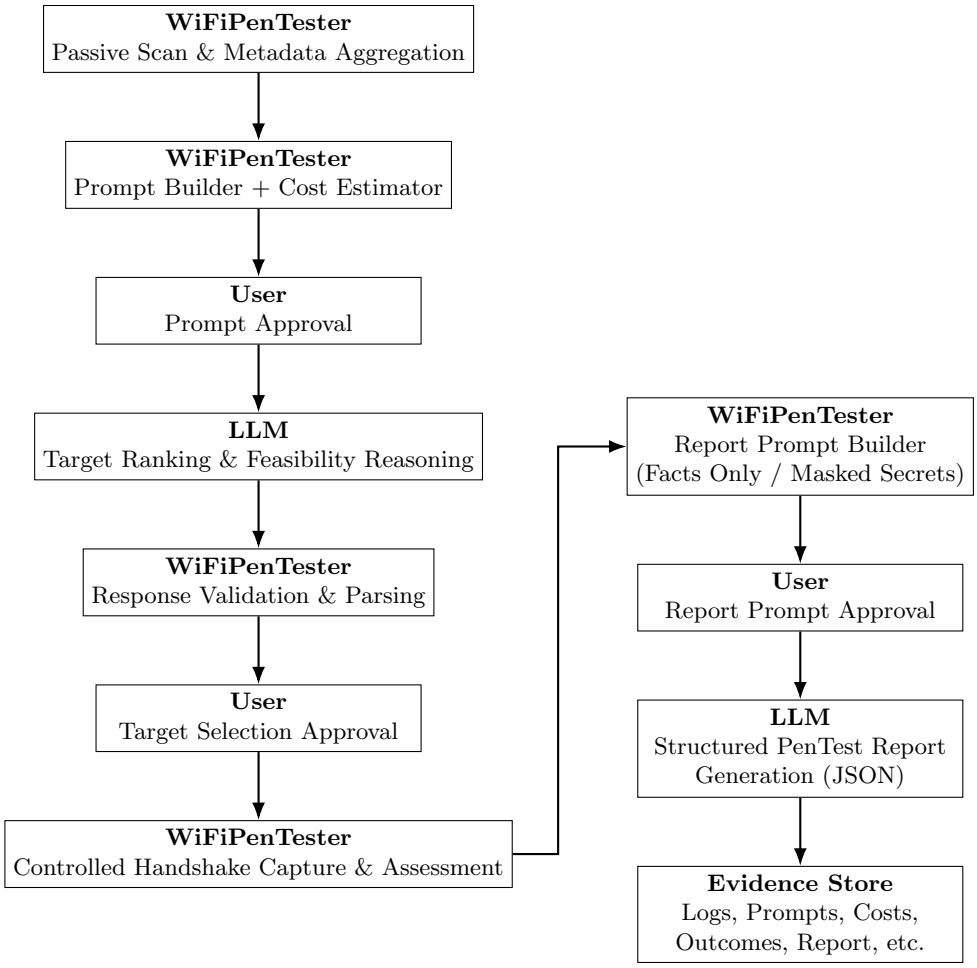

Fig. 1: Operational architecture of `WiFiPenTester`

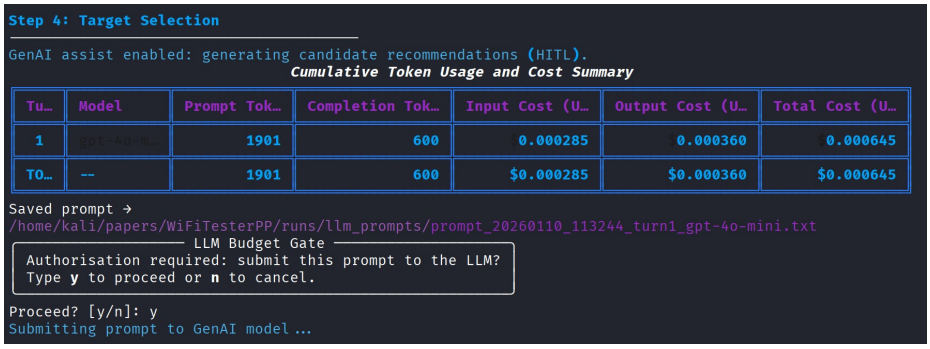

Fig. 2: Prompt token statistics and API cost estimation prior to LLM submission.

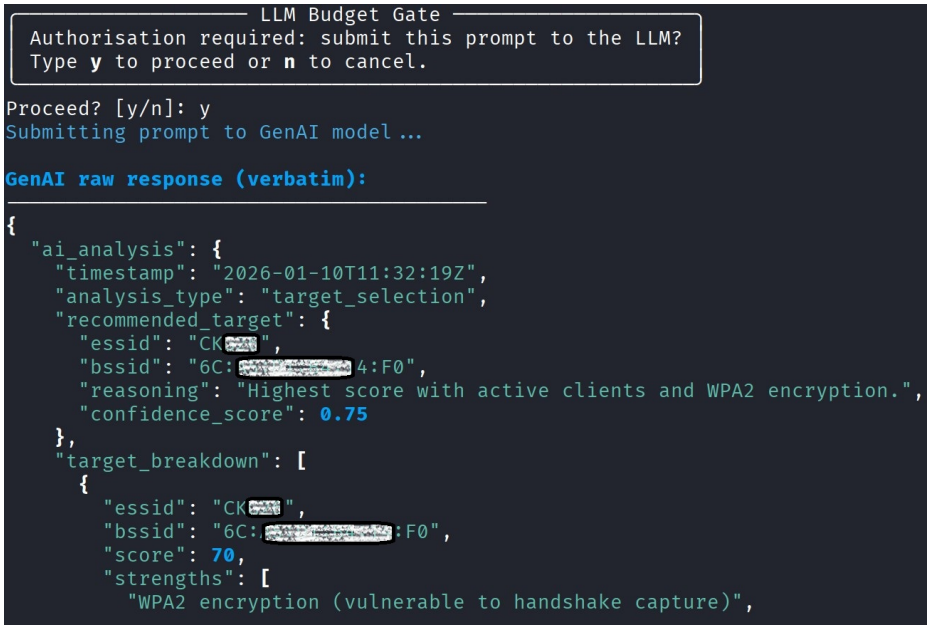

Fig. 3: LLM raw response.

Fig. 4: GenAI-based target ranking in structured JSON format.