Review



# EthiHack Pro v2.0: A Comprehensive Review of an Integrated, AI-Enhanced Ethical Hacking Toolkit

Anand Rawat *

Posted Date: 13 November 2025

doi: 10.20944/preprints202511.1000.v1

Keywords: ethical hacking; penetration testing; cybersecurity; software toolkit; python; Nmap; AI in security; automated reporting; vulnerability assessment

 doi:10.20944/preprints202511.1000.v1



*Review*

# EthiHack Pro v2.0: A Comprehensive Review of an Integrated, AI-Enhanced Ethical Hacking Toolkit

**Anand Rawat**

Dept of Computer Science and Engg, Pranveer Singh Institute Of Technology, (Affilated with AKTU), Kanpur, Uttar Pradesh, India; anandrawat138a@gmail.com

**Abstract**

The increasing sophistication of cyber threats necessitates a proactive approach to cybersecurity, with ethical hacking and penetration testing serving as foundational pillars of an effective defense strategy. However, the practical execution of a security assessment involves a fragmented workflow, often requiring practitioners to leverage numerous disparate command-line tools for reconnaissance, scanning, exploitation, and reporting. This paper presents a comprehensive review of EthiHack Pro v2.0, a hypothetical integrated software suite designed to address these challenges. The tool consolidates the entire penetration testing lifecycle into a single, GUI-driven platform built on Python and a robust ecosystem of security-focused libraries. We analyze its modular architecture, the technological stack underpinning its capabilities, and its structured workflow, which guides users from initial information gathering to final vulnerability exploitation. A significant innovation of the platform is its experimental AI-powered reporting module, which leverages generative AI to automate the creation of detailed, multi-audience security reports. This review examines the tool's design, functionality, and potential to streamline security assessments, making advanced testing methodologies more accessible and efficient for cybersecurity professionals.

**Index Terms:** ethical hacking; penetration testing; cybersecurity; software toolkit; python; Nmap; AI in security; automated reporting; vulnerability assessment

## I. Introduction

In the contemporary digital landscape, organizations face a persistent and evolving barrage of cyber threats. Consequently, a purely reactive security posture—waiting for a breach to occur before taking action—is no longer a viable strategy for protecting critical digital assets [1]. In response, the discipline of ethical hacking has transitioned from a niche, often misunderstood activity into a formalized and indispensable component of modern cybersecurity [2]. At its core, ethical hacking is the practice of utilizing hacking techniques in a lawful and authorized manner to identify and remediate security vulnerabilities within computer systems, networks, and applications [3]. Ethical hackers, also known as "white-hat hackers," operate with the explicit permission and intent to strengthen security, distinguishing them from their malicious counterparts ("black-hat hackers") who aim to cause harm [4]. This proactive "find and fix" strategy is often analogized to a rehearsal for a real-world cyberattack, allowing organizations to test their defenses in a controlled environment [1].

The execution of a comprehensive security assessment, such as a penetration test, is a multi-stage process encompassing reconnaissance, scanning, gaining access, maintaining access, and reporting [5]. Traditionally, this process requires security professionals to master and manually orchestrate a wide array of specialized command-line tools for each phase. This fragmentation can create significant workflow inefficiencies, introduce a steep learning curve for new practitioners, and complicate data aggregation for final reporting.

To address this challenge, integrated toolkits have been developed to consolidate these functions into a unified interface. This paper provides a detailed review of EthiHack Pro v2.0, a hypothetical, professional-grade software suite engineered to embody the principles of proactive defense and

 doi:10.20944/preprints202511.1000.v1



streamline the entire ethical hacking workflow. Built using the Python programming language, the tool provides a modular, graphical user interface (GUI) that guides a user through each phase of a penetration test, from initial reconnaissance to exploitation and final reporting.

A standout feature of EthiHack Pro v2.0 is its forward-looking integration of Artificial Intelligence (AI) to automate the most labor-intensive and critical phase of an engagement: report generation. This review will dissect the tool's architecture, its core technological components, its comprehensive workflow automation capabilities, its structured reporting framework, and its innovative AI-driven reporting engine. The objective is to provide a deep analysis of the tool's design and functionality, assessing its potential impact on the efficiency, accessibility, and quality of professional security testing.

## II. System Architecture and Technology Stack

The efficacy, maintainability, and scalability of a complex software tool are heavily dependent on its underlying architecture and technology choices. EthiHack Pro v2.0 is built upon a modular architecture and leverages the extensive Python ecosystem for its core functionalities, ensuring both robustness and flexibility.

### *A. Modular Architecture*

The application is intentionally designed using a modular architecture as opposed to a traditional, monolithic approach [6]. In a monolithic system, all components—user interface, business logic, and data access layers—are tightly integrated into a single, unified codebase. While initially simpler to develop, this model becomes unwieldy as the application grows, making updates and debugging complex and risky [7].

In contrast, EthiHack Pro v2.0's modular design breaks down the system into smaller, independent, and self-contained modules, each responsible for a specific piece of functionality (e.g., "Network Scanner," "Web Scraper," "Reporting") [6]. These modules communicate through well-defined interfaces, much like separate buildings on a university campus. This architectural choice offers several key advantages:

- **Easier Maintenance and Debugging**: If a bug appears in the "Network Scanner" module, developers can isolate and fix it without touching or risking the stability of other modules, simplifying the troubleshooting process significantly [7].
- **Improved Scalability and Flexibility**: New features, such as a module for cloud security scanning or IoT testing, can be developed and integrated without rewriting large parts of the existing codebase [8].
- **Parallel Development**: Different development teams can work on different modules simultaneously, which accelerates the overall development lifecycle and allows for specialization [7].

### *B. Core Technology Stack*

The tool is powered by a curated set of open-source Python libraries, chosen for their power, maturity, and relevance in the cybersecurity domain.

1. **User Interface (Tkinter)**: The Graphical User Interface (GUI) is constructed using Tkinter, Python's standard, built-in library for creating desktop applications [9]. As a wrapper for the mature and cross-platform Tcl/Tk GUI toolkit, Tkinter allows EthiHack Pro to run on Windows, macOS, and Linux without significant code changes, ensuring broad compatibility [10]. It provides the fundamental widgets—windows, buttons, text fields, and menus—that make the tool's complex backend capabilities accessible and manageable for the user [9].
2. **Network Interaction**: For network operations, the tool utilizes a trio of powerful and specialized libraries. The `python-nmap` library serves as a Python wrapper for the industry-standard Nmap engine, allowing the tool to programmatically define scan parameters, execute

 doi:10.20944/preprints202511.1000.v1



scans in the background, and parse the resulting XML output into a structured Python dictionary for easy processing [11]. For more granular and custom network tasks, Scapy is employed. Scapy is a powerful packet manipulation library that enables a developer to forge, send, capture, and decode network packets from scratch, building them layer by layer (e.g., Ethernet, IP, TCP), which is essential for crafting non-standard packets to test firewall rules or probe for obscure vulnerabilities [12]. All web-based interactions, from scraping to exploitation, are handled by the `Requests` library, the de facto standard for sending all types of HTTP requests in Python [13]. It capably manages headers, cookies, and sessions, making it indispensable for web-focused security testing.

3. **Data Parsing and Reporting**: Once data is retrieved, it must be parsed to extract valuable information. The tool integrates the `Beautiful Soup` library, which works in tandem with Requests to parse raw HTML and XML, transforming it into a navigable object tree that allows for the easy extraction of specific elements like links or employee names from a webpage [14]. For reconnaissance, the python-whois library is used to automate WHOIS queries and parse the results into a structured format, simplifying the collection of domain registration data [15]. The final deliverable of an engagement, the PDF report, is programmatically generated using the `reportlab` toolkit, an extensive library that provides precise, low-level control over the document's structure, text, images, and layout [16].
4. **Low-Level Networking**: For fundamental network communications where higher-level libraries are not suitable, EthiHack Pro utilizes Python's built-in `socket` module. This library provides the basic interface for creating TCP and UDP connections, serving as the foundation upon which libraries like Requests are built [17]. It is used when direct control over a raw data stream is required to interact with a non-standard service.

## III. The Penetration Testing Workflow

EthiHack Pro v2.0 is meticulously structured to mirror the logical phases of a professional penetration test, providing dedicated, interconnected modules for each stage of the engagement [5]. This guided workflow ensures a systematic and comprehensive assessment.

### *A. Phase 1: Reconnaissance (Footprinting)*

This initial and most critical phase focuses on intelligence gathering to create a comprehensive map of the target's digital footprint [18]. The quality of information gathered here directly influences the success of all subsequent phases [19]. The tool automates both passive and active reconnaissance techniques.

- **Passive Reconnaissance (OSINT)**: This involves gathering information without directly interacting with the target's systems, making the activities virtually undetectable [18]. EthiHack Pro's OSINT modules automate the collection of data from public sources [20]. This includes scraping the company's official website for employee names, analyzing social media platforms like LinkedIn for roles and hierarchies, and examining job postings, which can inadvertently reveal the technologies and software versions an organization uses [21].
- **WHOIS Lookups**: The tool includes a "Domain Analyzer" that automates WHOIS lookups. The WHOIS system is a public database containing registration information for domain names [22]. A query can uncover the registrant's name and contact details, the authoritative DNS name servers for the domain, and domain registration and expiration dates, providing valuable leads for both technical and social engineering attacks [23].
- **Active Reconnaissance (DNS Enumeration)**: In contrast to passive methods, active reconnaissance involves direct interaction with the target's infrastructure [18]. The tool's "DNS Mapper" performs DNS enumeration, systematically querying a target's DNS servers to translate domain names into IP addresses [24]. It queries for various record types to build a network map: an **A record** maps a hostname (e.g., `vpn`) to an IPv4 address; an **MX record** identifies mail

 doi:10.20944/preprints202511.1000.v1



servers; an **NS record** points to the authoritative name servers; and a **TXT record** can contain miscellaneous information, sometimes leaking details about security configurations or third-party services [24].

*B. Phase 2: Scanning*

Following reconnaissance, the scanning phase uses the gathered intelligence to probe target systems directly, aiming to identify live hosts, open ports, and the specific services running on them [25].

- **Port Scanning with Nmap**: At the core of this phase is the "Network Scanner" module, which is powered by the Nmap engine [11]. To understand this, one can use the analogy of a server being an office building with an IP address as its street address; the ports are the numbered offices inside (from 0 to 65535), each providing a specific service [26]. For example, web traffic typically goes to port 80 (HTTP) or 443 (HTTPS). An open port has an application actively listening for connections and is a potential gateway for an attacker [27]. The scanner sends crafted packets to determine a port's state: **Open** (an application is listening), **Closed** (no application is listening), or **Filtered** (a firewall is blocking access) [28].
- **Service and Version Detection**: Identifying an open port is only the first step; knowing the specific software and version on that port is far more critical [28]. The scanner performs service and version detection by sending probes that elicit unique responses from applications, often capturing a "banner" that explicitly states the software name and version (e.g., `Apache/2.4.41`). An ethical hacker can then cross-reference this version with public vulnerability databases (like CVE) to find known, exploitable flaws.
- **Vulnerability Scanning with NSE**: To streamline this process, the tool utilizes the Nmap Scripting Engine (NSE), a feature that allows for the automation of a wide variety of networking tasks using scripts written in the Lua language [29]. EthiHack Pro deploys curated NSE scripts based on the services discovered. This includes scripts from the `vuln` category, which specifically check for known vulnerabilities, transforming the network scanner into an automated vulnerability identification tool [29].

*C. Phase 3: Gaining Access (Exploitation)*

This is the most active stage, where the theoretical vulnerabilities discovered are put to the test to demonstrate their real-world impact [30]. EthiHack Pro's "Exploitation Suite" provides modules to target common, high-impact vulnerabilities.

- **Web Application Attacks (OWASP Top 10)**: The modules are heavily aligned with the OWASP Top 10, a consensus document on the most critical web application security risks [31].
  - **SQL Injection (SQLi)**: This attack injects malicious SQL code into an application's input fields [32]. The tool's "SQLi Module" automates sending payloads like `' OR 1=1--` to login forms or search bars. If the application is vulnerable, this can bypass authentication by creating a universally true logical condition, allowing an attacker to read, modify, or delete data from the database [33].
  - **Cross-Site Scripting (XSS)**: This attack targets other users of an application by injecting malicious scripts into content that is then displayed in their browsers [34]. The tool tests for both **Reflected XSS**, where the script is part of a malicious link the victim must click, and **Stored XSS**, where the script is permanently saved on the server (e.g., in a comment) and affects any user who views the page [34].
  - **Local File Inclusion (LFI)**: This vulnerability allows an attacker to include files on the server by manipulating URL parameters [35]. The tool tests this by using directory traversal sequences (`../`) in URL parameters to try and break out of the web root directory and access sensitive system files, such as `/etc/passwd` on a Linux server [35].

 doi:10.20944/preprints202511.1000.v1



- **Network and Infrastructure Attacks**: The suite also includes tools for infrastructure testing. The "Directory Buster" module uses large dictionary files ("wordlists") to brute-force the names of hidden files and directories on a web server, looking for administrative portals or forgotten backup files by checking for `200 OK` HTTP responses [36].

*D. Phase 4: Post-Exploitation and Analysis*

Gaining initial access is often just the beginning [30]. This phase explores what an attacker could do after a successful breach.

- **Maintaining Access**: The objective is to establish persistence—the ability to maintain access even if the system is rebooted or the initial vulnerability is patched [37]. The tool includes functionalities to test for weaknesses that would allow an attacker to install backdoors, create hidden user accounts, or deploy Remote Access Tools (RATs) [37].
- **Log Analysis and Covering Tracks**: Every action on a system generates logs, which are digital footprints of activity [38]. A sophisticated attacker will attempt to cover their tracks by deleting or modifying log files to evade detection [37]. EthiHack Pro includes a "Log Analyzer" module that works from two perspectives. It can be used by the ethical hacker to test an organization's ability to detect log tampering. From the defender's perspective, the same module can parse large volumes of log data to identify anomalies and indicators of compromise, such as multiple failed login attempts from a single IP, logins at unusual times, or unusual data access patterns [38].
- **Pivoting**: This is the technique of using a compromised system to attack other systems on the same internal network [37]. Often, internal systems are less secure and not directly accessible from the internet. An ethical hacker can use a compromised public-facing web server as a "pivot point" to launch scans and attacks against internal assets like domain controllers or databases, demonstrating how a single external vulnerability can lead to a full internal network compromise [37].

## IV. The Professional Reporting Framework

The culmination of a penetration testing engagement is not the exploit itself, but the delivery of the final report. This document is the single most important outcome, translating complex technical findings into actionable business intelligence [39]. A successful test that produces a poor report is a failure, as it provides no value to the client. EthiHack Pro v2.0's reporting engine is designed to produce professional reports structured to communicate effectively with the two distinct audiences within a client's organization: non-technical executives and hands-on technical staff [40].

*A. Key Components of the Report*

1. **Executive Summary**: This is the first and most crucial section for non-technical stakeholders like C-level executives and board members [40]. It provides a high-level, concise overview of the engagement's findings in plain, accessible language, avoiding technical jargon. It focuses on the business context and potential impact, summarizing the key findings categorized by severity, assessing the potential business impact in terms of financial loss or reputational damage, and providing strategic recommendations [39].
2. **Methodology and Scope**: This section establishes the credibility and transparency of the test [40]. It clearly defines the scope, listing all systems, applications, and IP ranges that were tested, as well as those that were explicitly out-of-scope. It also details the methodology used (e.g., black-box, white-box) and the testing frameworks followed, such as the OWASP Web Security Testing Guide (WSTG) [40].
3. **Technical Findings**: This is the core of the report for the technical audience, such as developers and system administrators [39]. It provides a detailed, vulnerability-by-vulnerability breakdown. Each finding includes a clear description of the weakness, a **Proof-of-Concept (PoC)**

 doi:10.20944/preprints202511.1000.v1



with concrete evidence and steps to reproduce the exploit (often with screenshots or code snippets), an analysis of the potential impact, and a risk rating [40].

4. **Remediation Recommendations**: A report that only identifies problems is of limited value. This section provides clear, specific, and actionable guidance on how to fix each identified vulnerability [39]. Instead of generic advice, it offers detailed options tailored to the client's environment, such as specific patches to apply, configuration changes to implement, or code examples to fix a flaw [40].

*B. Quantifying Risk with CVSS*

To provide a consistent and objective measure of a vulnerability's severity, the reports generated by EthiHack Pro utilize the Common Vulnerability Scoring System (CVSS) [41]. CVSS is an open framework that produces a numerical score from 0.0 to 10.0, allowing organizations to prioritize remediation efforts in a vendor-agnostic manner [42]. The CVSS Base Score, which represents the intrinsic qualities of a vulnerability, is calculated from several key metrics:

- **Attack Vector (AV)**: Describes how the vulnerability can be exploited (e.g., over the Network vs. Physical access) [42].
- **Attack Complexity (AC)**: Measures the difficulty of the attack (Low vs. High complexity) [42].
- **Privileges Required (PR)**: The privilege level an attacker needs beforehand (e.g., None vs. High/administrative) [42].
- **User Interaction (UI)**: Whether a user must take an action, like clicking a link (Required vs. None) [42].
- Impact (Confidentiality, Integrity, Availability): Measures the potential loss of data confidentiality, integrity, or service availability [42].

This numerical score is mapped to a qualitative rating (e.g., Critical, High, Medium, Low) to provide a clear, data-driven basis for risk assessment [41].

## V. Innovation Spotlight: AI-Driven Reporting

While a structured report is critical, its manual creation is one of the most time-consuming and labor-intensive phases for a security professional [43]. EthiHack Pro v2.0 introduces an experimental "AI Reporting" module designed to leverage generative AI to automate and revolutionize this process. This functionality builds upon the broader trend of using AI in cybersecurity for tasks like automated threat detection and vulnerability management [44].

The AI reporting engine works through a multi-step process:

1. **Data Ingestion and Structuring**: The model ingests all structured and unstructured data collected during the test, including Nmap scan results, vulnerability findings, log files, and notes from the human tester [43].
2. **Analysis and Correlation**: The AI analyzes the aggregated data to identify patterns. For example, it can connect a discovered outdated software version with a specific CVE identifier and a publicly available exploit, automatically assessing the severity and potential impact [43].
3. **Content Generation for Multiple Audiences**: This is the core generative function. Trained on thousands of security reports, the model generates human-readable text tailored to different sections of the report [45]. For the **Executive Summary**, it can synthesize the highest-risk findings and describe their potential business and financial impact in non-technical language. For the **Technical Findings**, it can generate a detailed, technically accurate description of a vulnerability like SQL injection. For the **Remediation Section**, it can provide code-specific guidance, suggesting the exact code changes or configuration updates needed to fix the vulnerability [43].

It is crucial to note that this feature operates on a "human-in-the-loop" model. The AI generates a draft report, which is then reviewed, edited, and validated by the human expert to ensure accuracy,

 doi:10.20944/preprints202511.1000.v1



context, and quality [45]. This hybrid approach combines the speed and scale of AI with the critical thinking of a skilled professional, aiming to dramatically increase efficiency while improving the consistency of the final deliverable.

## VI. Discussion and Future Directions

Beyond its core workflow, EthiHack Pro v2.0 is engineered with a notable focus on robustness, usability, and adherence to modern software development best practices. The implementation of robust error handling is a key example. By using specific `try...except` blocks to catch distinct exceptions like `requests.exceptions.Timeout` or `requests.exceptions.ConnectionError`, the tool can provide precise feedback to the user and handle unpredictable network environments gracefully instead of crashing [46]. This makes the application more reliable and user-friendly, especially during long-running scans.

Furthermore, the tool's design promotes adaptability through user-configurable settings managed via external configuration files (e.g., in `.ini` or `.json` format). This is a significant advantage over hardcoded parameters, allowing practitioners to easily tailor the tool to the specific needs of an engagement [47]. Users can change target IP ranges, specify custom wordlists for directory busting, adjust Nmap scan intensity, or select different report templates without altering the source code. This enhances flexibility and makes the tool more accessible to users who may not be programmers.

The inclusion of a `requirements.txt` file for dependency management is another critical feature for ensuring reproducibility and collaboration [48]. This file allows any developer to recreate the exact software environment by installing the precise versions of all necessary libraries with a single command, eliminating version incompatibility issues [48].

For future development, the tool's modular architecture provides a strong foundation for expansion [8]. New modules could be developed to address emerging attack surfaces such as cloud infrastructure security (e.g., for AWS, Azure, GCP) and the security of Internet of Things (IoT) devices. The AI capabilities could also be extended beyond reporting. An advanced AI could potentially assist in other phases of the test, such as by analyzing reconnaissance data to suggest high-priority targets or by helping to chain multiple low-risk vulnerabilities into a high-impact exploit path [45].

## VII. Conclusion

EthiHack Pro v2.0, as a conceptual model, represents a significant step toward creating a truly integrated, efficient, and accessible ethical hacking toolkit. By consolidating the disparate phases of a penetration test into a single, GUI-driven platform, it effectively addresses the workflow fragmentation and inefficiencies that are common challenges for cybersecurity professionals. Its modular architecture and reliance on a powerful and mature Python technology stack provide a flexible and scalable foundation, ensuring the tool can evolve alongside the threat landscape.

The tool's most significant and forward-looking contribution is its exploration of AI-driven report generation. This feature has the potential to revolutionize how security findings are documented and communicated, a critical yet often time-consuming aspect of security assessments. By automating the laborious task of report writing, it allows security experts to dedicate more of their valuable time to the complex, creative, and critical-thinking tasks of vulnerability discovery and analysis. While hypothetical, EthiHack Pro v2.0 serves as a compelling blueprint for the next generation of security assessment tools, where deep workflow integration and intelligent automation will be paramount in the ongoing effort to secure digital infrastructure.

 doi:10.20944/preprints202511.1000.v1



## References


1. NetCom Learning, "What Is Ethical Hacking? A Beginner's Guide to Cybersecurity," NetCom Learning Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.netcomlearning.com/blog/what-is-ethical-hacking
2. IBM, "What is Ethical Hacking?," IBM, Accessed: Sep. 27, 2025. [Online]. Available: https://www.ibm.com/think/topics/ethical-hacking
3. EC-Council, "Ethical Hacking for Beginners and Top Career Paths in Cybersecurity," EC-Council, Accessed: Sep. 27, 2025. [Online]. Available: https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/ethical-hacking-for-beginners/
4. Coursera, "What Is Ethical Hacking? Definition and Examples," Accessed: Sep. 27, 2025. [Online]. Available: https://www.coursera.org/articles/what-is-ethical-hacking
5. EC-Council, "Learn About The Five Penetration Testing Phases," EC-Council, Accessed: Sep. 27, 2025. [Online]. Available: https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/
6. R. Lin, "Monolithic vs Modular," Medium, Accessed: Sep. 27, 2025. [Online]. Available: https://medium.com/@berto168/monolithic-vs-modular-9b6d69684a2c
7. Harrison Clarke, "Benefits of Modular Architecture: Moving from Monolithic to Modular," Harrison Clarke Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.harrisonclarke.com/blog/benefits-of-modular-architecture-moving-from-monolithic-to-modular
8. Selleo, "How Does Modular Software Architecture Improve Scalability?" Selleo Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://selleo.com/blog/how-does-modular-software-architecture-improve-scalability
9. GeeksforGeeks, "Python Tkinter," GeeksforGeeks, Accessed: Sep. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org/python/python-gui-tkinter/
10. Wikipedia, "Tkinter," Accessed: Sep. 27, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Tkinter
11. C. Robert, "How to Automate Nmap with Python — A Beginner-Friendly Guide," Medium, Accessed: Sep. 27, 2025. [Online]. Available: https://medium.com/@carylrobert16/how-to-automate-nmap-with-python-a-beginner-friendly-guide-a0614dd06950
12. Scapy.net, "Introduction — Scapy 2.6.1 documentation," Accessed: Sep. 27, 2025. [Online]. Available: https://scapy.readthedocs.io/en/latest/introduction.html
13. Real Python, "Python's Requests Library (Guide)," Real Python, Accessed: Sep. 27, 2025. [Online]. Available: https://realpython.com/python-requests/
14. Real Python, "Beautiful Soup: Build a Web Scraper With Python," Accessed: Sep. 27, 2025. [Online]. Available: https://realpython.com/beautiful-soup-web-scraper-python/
15. E. Willians, "Building a Robust Domain Checker with DNS and WHOIS in Python," DEV Community, Accessed: Sep. 27, 2025. [Online]. Available: https://dev.to/ericsonwillians/building-a-robust-domain-checker-with-dns-and-whois-in-python-2e7m
16. ReportLab, "ReportLab PDF Library User Guide," ReportLab Inc., Accessed: Sep. 27, 2025. [Online]. Available: https://www.reportlab.com/docs/reportlab-userguide.pdf
17. Python Software Foundation, "Socket Programming HOWTO — Python 3.13.7 documentation," Python Docs, Accessed: Sep. 27, 2025. [Online]. Available: https://docs.python.org/3/howto/sockets.html
18. GeeksforGeeks, "Reconnaissance - Penetration Testing," GeeksforGeeks, Accessed: Sep. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org/software-engineering/reconnaissance-penetration-testing/
19. Strikegraph, "Pen Testing Phases & Steps," Accessed: Sep. 27, 2025. [Online]. Available: https://www.strikegraph.com/blog/pen-testing-phases-steps
20. Imperva, "Open-Source Intelligence (OSINT)," Imperva Learning Center, Accessed: Sep. 27, 2025. [Online]. Available: https://www.imperva.com/learn/application-security/open-source-intelligence-osint/
21. BitSight Technologies, "How to Use the OSINT Framework: Sources, Tools, & Steps," BitSight, Accessed: Sep. 27, 2025. [Online]. Available: https://www.bitsight.com/learn/cti/osint-framework
22. GeeksforGeeks, "What is Whois Footprinting?," GeeksforGeeks, Accessed: Sep. 27, 2025. [Online]. Available: https://www.geeksforgeeks.org/ethical-hacking/what-is-whois-footprinting/

 doi:10.20944/preprints202511.1000.v1




23. WhoisFreaks, "Mastering WHOIS OSINT for Effective Domain and IP Investigations," WhoisFreaks Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://whoisfreaks.com/resources/blog/mastering-whois-osint-for-effective-domain-and-ip-investigations
24. Recorded Future, "What is DNS Enumeration? Top Tools and Techniques Explained," Recorded Future, Accessed: Sep. 27, 2025. [Online]. Available: https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/dns-enumeration
25. Vertex Cyber Security, "Scanning In Penetration Testing - An Introduction," Vertex Cyber Security, Accessed: Sep. 27, 2025. [Online]. Available: https://www.vertexcybersecurity.com.au/scanning-in-penetration-testing/
26. NordVPN, "What are open ports? Risks and security," NordVPN Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://nordvpn.com/blog/what-are-open-ports/
27. BeyondTrust, "What is an Open Port & What are the Security Implications?," BeyondTrust Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.beyondtrust.com/blog/entry/what-is-an-open-port-what-are-the-security-implications
28. Nmap.org, "Chapter 15. Nmap Reference Guide," Nmap Network Scanning, Accessed: Sep. 27, 2025. [Online]. Available: https://nmap.org/book/man.html
29. Nmap.org, "Nmap Scripting Engine (NSE)," Accessed: Sep. 27, 2025. [Online]. Available: https://nmap.org/book/man-nse.html
30. Pentest-Standard, "Exploitation," Read the Docs, Accessed: Sep. 27, 2025. [Online]. Available: https://pentest-standard.readthedocs.io/en/latest/exploitation.html
31. OWASP Foundation, "OWASP Top Ten," OWASP, Accessed: Sep. 27, 2025. [Online]. Available: https://owasp.org/www-project-top-ten/
32. OWASP Foundation, "SQL Injection," OWASP Community, Accessed: Sep. 27, 2025. [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection
33. Fortinet, "What is SQL Injection?," Fortinet, Accessed: Sep. 27, 2025. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/sql-injection
34. PortSwigger, "What is cross-site scripting (XSS) and how to prevent it?," Web Security Academy, Accessed: Sep. 27, 2025. [Online]. Available: https://portswigger.net/web-security/cross-site-scripting
35. Bright Security, "Local File Inclusion (LFI): Understanding and Preventing LFI Attacks," Bright Security Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://brightsec.com/blog/local-file-inclusion-lfi/
36. Sprocket Security, "Directory Brute-forcing at Scale," Sprocket Security Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.sprocketsecurity.com/blog/directory-brute-forcing-at-scale
37. NetSPI, "The Penetration Testing Life Cycle Explained," NetSPI Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.netspi.com/blog/executive-blog/penetration-testing-as-a-service/the-penetration-testing-life-cycle-explained/
38. Splunk, "Log Analysis: A Complete Introduction," Splunk Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.splunk.com/en_us/blog/learn/log-analysis.html
39. Strobes Security, "Essential Elements of a Penetration Testing Report," Strobes Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://strobes.co/blog/penetration-testing-report-key-elements-you-cant-miss/
40. Rhino Security Labs, "4 Things Every Penetration Test Report Should Have," Rhino Security Labs, Accessed: Sep. 27, 2025. [Online]. Available: https://rhinosecuritylabs.com/penetration-testing/four-things-every-penetration-test-report/
41. SANS Institute, "What is CVSS - Common Vulnerability Scoring System," SANS Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.sans.org/blog/what-is-cvss
42. National Institute of Standards and Technology, "A Complete Guide to the Common Vulnerability Scoring System (CVSS)," NIST, Publication ID 51198, Accessed: Sep. 27, 2025. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198
43. Global App Testing, "Generative AI in Penetration Testing - The Comprehensive Guide," GAT Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.globalapptesting.com/blog/generative-ai-penetration-testing

 doi:10.20944/preprints202511.1000.v1




44. Microsoft Security, "What Is AI for Cybersecurity?," Microsoft, Accessed: Sep. 27, 2025. [Online]. Available: https://www.microsoft.com/en-us/security/business/security-101/what-is-ai-for-cybersecurity
45. Aikido Security, "Using Generative AI for Pentesting: What It Can (and Can't) Do," Aikido Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.aikido.dev/blog/ai-for-pentesting
46. Honeybadger, "The ultimate guide to Python exception handling," Honeybadger Developer Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.honeybadger.io/blog/a-guide-to-exception-handling-in-python/
47. UpGuard, "What Is Configuration Management and Why Is It Important?," UpGuard Blog, Accessed: Sep. 27, 2025. [Online]. Available: https://www.upguard.com/blog/5-configuration-management-boss
48. freeCodeCamp, "How to Create and Pip Install Requirements.txt in Python," freeCodeCamp News, Accessed: Sep. 27, 2025. [Online]. Available: https://www.freecodecamp.org/news/python-requirementstxt-explained/