Neo Hire | Societe Generale | 2026 RVCE Hackathon

Network Pen Testing Automation Workflow using LLM

Introduction & System Overview

In today's rapidly evolving digital landscape, organizations are constantly exposed to a wide range of cyber threats. Manual penetration testing, while thorough, often struggles to keep pace with these threats due to time and resource constraints. This project addresses that challenge by presenting an **automated network pentesting workflow**, integrating **Large Language Models (LLMs)** and workflow automation tools like **n8n**.

The primary objective is to automate various stages of the pentesting lifecycle, including reconnaissance, vulnerability assessment, analysis, and reporting. By leveraging LLMs such as ChatGPT, we can enhance intelligence at each step—enabling rapid analysis, contextual risk understanding, and clear, actionable reporting.

Key Features:

- End-to-end automated pentest execution using n8n
- LLM-assisted parsing, risk analysis, and reporting
- Modular and extensible architecture
- Simple interface and user-triggered scans

Technologies Used:

- **n8n** – Workflow orchestration
- **ChatGPT** – NLP-powered parsing and report generation
- **Nessus** – Network reconnaissance and port scanning
- **Nessus (optional)** – Advanced vulnerability scanning
- **Email/Slack/File** – Output distribution

Workflow Components

1. Webhook Trigger

This component serves as the entry point for the entire pentesting workflow. It can be triggered by a user via a simple frontend (e.g., an HTML form), which collects the target IP/domain and sends it to an n8n webhook over HTTPS.

2. Execute Command: Nessus

Once the webhook is triggered, an Execute Command node in n8n runs a predefined **Nessus scan** command. The scan can be customized based on desired intensity, such as:

Nessus -A -T4 <target>

The output from Nessus is captured in text format for further parsing.

3. Parse Output / LLM Parser

The raw output from Nessus is passed to ChatGPT (or another LLM) via a prompt designed to:

- Extract open ports, services, and versions
- Identify potential vulnerabilities or misconfigurations
- Suggest preliminary risk levels

This enables faster, more intelligent analysis compared to manual parsing.

4. Risk Analysis or Nessus API (Optional)

For deeper vulnerability insights, the workflow can optionally query the **Nessus API** to scan the same target and retrieve CVE-based vulnerability data. This data can then be passed back to the LLM for enhanced analysis and prioritized risk evaluation.

5. LLM Report Generator

After all findings are parsed and analyzed, the LLM generates a **brief, structured report** that includes:

- A summary of findings

- Risk categorization (Low/Medium/High)
- Recommended remediation steps
- Additional context if available (e.g., CVE links)

This report is crafted in natural language and is suitable for technical teams and management alike.

6. Output Channels (Email/Slack/File)

Finally, the report is delivered to stakeholders through one or more of the following methods:

- **Email** – Sends a formatted report as an attachment or inline text
- **Slack** – Posts key findings to a security channel
- **File Output** – Stores the report on disk or uploads to a shared drive/cloud bucket

---

LLM Integration

Role of LLM in the Workflow

LLMs are embedded at two key stages:

- **Parsing Nessus output**: Extract structured insights from unstructured CLI output
- **Generating the report**: Transform findings into human-readable, actionable content

Prompt Design Strategy

Prompts are carefully engineered to:

- Minimize hallucination
- Maintain consistency in output format
- Guide the LLM to use markdown-style structure (for readability)
- Encourage inclusion of CVE references and standard terminology

Example Parsing Prompt:

"""

Below is the result of an Nessus scan. Please extract open ports, service details, and any security-related observations. Suggest possible vulnerabilities and remediation guidance if applicable. Format the output as a structured report.

Setup and Deployment

Prerequisites

- Docker (optional for easy deployment)
- n8n installed (cloud or self-hosted)
- OpenAI API Key (or compatible LLM endpoint)
- Nessus installed on the machine running n8n
- (Optional) Nessus and API token

Installation Steps

1. **Set up n8n** instance (cloud or self-hosted)
2. **Create webhook** trigger node
3. **Add Execute Command** node to run Nessus
4. **Use Function Node** to format Nessus output
5. **Call OpenAI API** using HTTP Request node
6. (Optional) **Query Nessus API** and merge data
7. **Generate Report** using second OpenAI call
8. **Deliver Output** using Email/Slack/File node

Configuration Notes

- Ensure proper timeout settings for long Nessus scans
- Secure your webhook endpoint
- Store API keys in environment variables or n8n credentials

Usage Guide

1. **User submits target IP/domain** via web form or direct webhook call
2. n8n triggers the automated scan
3. Nessus results are analyzed by LLM
4. (Optional) Nessus data enriches the scan results
5. LLM generates and formats a pentest report
6. Report is automatically sent via configured channels

Example Run:

- Target: 192.168.1.1
- Nessus Output: 3 open ports (22, 80, 443)
- LLM Report: Indicates outdated Apache on port 80, SSH version disclosure on port 22
- Output: Report sent to email and stored as PDF

Sample Report

Target: 192.168.1.1

Scan Date: 2025-07-13

Findings:

- Port 22 (SSH): Open, OpenSSH 7.2 detected. Version disclosure; consider upgrade.

- Port 80 (HTTP): Apache 2.4.29 detected. Known vulnerabilities: CVE-2021-41773.

- Port 443 (HTTPS): TLS 1.0 supported. Recommend enforcing TLS 1.2+

Risk Assessment:

- High Risk: Apache Vulnerability (CVE-2021-41773)

- Medium Risk: Weak TLS configuration

- Low Risk: SSH Version Disclosure

Remediation Suggestions:

- Patch Apache server immediately
- Update SSH and disable version banners
- Enforce TLS 1.2 or higher

Extensibility & Customization

The workflow is modular and designed for future growth:

- Add support for tools like Shodan, OpenVAS, or Burp Suite
- Replace Nessus with masscan for faster scans
- Expand prompt library for different scan types
- Add authentication for protected APIs (e.g., Google Chat, ServiceNow)
- Export reports in various formats (Markdown, PDF, DOCX)

Appendix

Example Prompts

- Nessus Parsing Prompt
- Nessus Result Summary Prompt
- Report Structuring Prompt

Glossary

- **LLM**: Large Language Model
- **n8n**: Node-based workflow automation tool
- **CVE**: Common Vulnerabilities and Exposures
- **Nessus**: Network Mapper
- **Nessus**: Vulnerability Assessment Tool

References

- Nessus Documentation
- OpenAI API Reference
- n8n Documentation
- Nessus API Guide