

# Self-Hosted Workflow Automation for AI-Based Cybersecurity Operation

Hareign Casaclang  
Dept. of Computer Science  
University of Nevada, Las Vegas  
Las Vegas, NV, USA  
casaclang@unlv.nevada.edu  
0009-0000-4740-4035

Bianca Ionescu  
Dept. of Cybersecurity  
University of Nevada, Las Vegas  
Las Vegas, NV, USA  
ionesb1@unlv.nevada.edu  
0009-0002-7395-9565

Yoohwan Kim  
Dept. of Computer Science  
University of Nevada, Las Vegas  
Las Vegas, NV, USA  
yoohwan.kim@unlv.edu  
0000-0002-7321-9527

Ju-Yeon Jo  
Dept. of Computer Science  
University of Nevada, Las Vegas  
Las Vegas, NV, USA  
juyeon.jo@unlv.edu  
0000-0001-6524-9825

**Abstract**—Cybersecurity operations often involve repetitive tasks such as running Nmap scans, analyzing logs, and performing Open-Source Intelligence (OSINT) investigations. These processes are essential for maintaining security but consume time and resources that many small organizations cannot spare. While commercial automation platforms exist to reduce this workload, they are typically costly and inaccessible to businesses without dedicated IT staff. This paper investigates n8n, a self-hosted and low-cost workflow automation platform, as a practical alternative for cybersecurity automation. By integrating security tools and external large language models (LLMs) such as ChatGPT, Gemini, and Ollama, n8n can automate vulnerability scanning, assign severity ratings, and generate reports tailored to both technical and executive stakeholders. Experiments show that n8n workflows can effectively combine traditional scans with Artificial Intelligence (AI)-driven analysis to produce actionable outputs. Although limitations remain, including a steep learning curve and restrictions in the free tier, n8n demonstrates potential for broadening access to automation in cybersecurity. For small organizations, this approach provides a cost-effective way to strengthen security posture, while in academic contexts it provides a hands-on platform for teaching and experimenting with automation and AI in cybersecurity.

**Keywords**—Cybersecurity, Workflow Automation, n8n, Nmap, Artificial Intelligence (AI), AI Agent, Vulnerability scanning

## I. INTRODUCTION

Within cybersecurity, there are many tedious tasks that take a large amount of time and resources. Tasks such as gathering intelligence, log analysis, and reporting are essential, but take numerous hours to complete. When it comes to automating cybersecurity tasks, a structured approach is necessary to ensure that workflows are not only functional but also adaptable, reliable, and secure. Due to this, task automation is becoming necessary to reduce human errors and improve overall efficiency. A self-hosted workflow automation platform such as n8n offers flexibility and control that is specifically valuable in AI-driven cybersecurity operations.

n8n was chosen as the main platform for this research due to how recently it was created, its integrations, as well as how one can host it on their local machine without paying a monthly subscription. Created in October 2019, n8n can integrate data and functions between more than 200 established applications. There is a cloud-based version of n8n that can be used for managed hosting; however, this requires a monthly subscription after a fourteen-day trial, with a limit of how many executions can be run. n8n has been seen as a technical product for most of its duration, but according to the founder, Jan Oberhauser, in an interview he mentioned that n8n's plan is to make the platform simpler to use and easier to adopt. This way not only can users with technical knowledge use n8n, but even those, such as a marketing team, who do not have a technical background can utilize n8n [1]. n8n can empower "citizen developers" or non-technical staff to innovate within their organization due to it being a low-code/no-code platform [2].

Automated reporting tools can generate comprehensive and easily interpretable reports making it easier for stakeholders to understand the security landscape [3]. AI-driven cybersecurity solutions can automate routine tasks, optimize resource allocation, and focus on strategic decision-making [4]. This research will focus on how to set up one's self-hosted n8n instance, understanding how to use AI agents, and creating a workflow that analyzes command input to create two types of vulnerability reports: technical and non-technical.

## II. RELATED WORKS

Research into workflow automation and AI-driven orchestration has expanded into multiple domains, with several works addressing the potential and limits of platforms like n8n. Puiu demonstrates how n8n can be leveraged in cybersecurity contexts to triage and process security reports using large language models [5]. This study highlights the orchestration role of n8n, where LLMs handle text-heavy tasks such as report summarization and classification, while workflows manage data routing, validation, and escalation. The work confirms that n8n is not only suitable for general automation but also adaptable to security-specific applications where AI can reduce analyst workload and improve incident response efficiency.

TABLE I. Workflow Automation Tools Comparison.

Tool	Capabilities	AI Integration	Cost	Cybersecurity Applicability
n8n	Open-source, general-purpose; API and service integration; self-hostable	External models setup	Free / Paid Subscription	Flexible, low-cost; lacks native SOC governance
SOAR Platforms	Purpose-built for SOCs; prebuilt security integrations, case management, audit logs	Increasing AI for enrichment / triage	Enterprise licensing	Enterprise-level, scalable, compliance-ready
Power Automate	LCNC for Microsoft ecosystem	Azure AI, Copilot (vendor-locked)	Enterprise bundle	Easy to use; limited custom security workflows
Opal AI	Experimental LCNC	Google Gemini integrated	No standard pricing	Accessible, but lacks governance & resilience

Modern low-code/no-code platforms such as Microsoft Power Automate and Google's Opal AI aim to lower the barrier to automation by providing natural language workflows and vendor-specific integrations. Gupta *et al.* highlights the accessibility of Power Automate but notes the need to reduce the steep learning curve for non-technical users [6]. In contrast, n8n emphasizes adoptability, extensibility, and self-hosting [1], putting it between general-purpose low-code/no-code tools and enterprise level SOAR platforms.

These works illustrate the dual nature of n8n: versatile enough to support advanced cybersecurity workflows, as shown by Puiu [5], yet subject to structural limitations that have also been observed in business contexts, as Cunha discusses accounting, highlights n8n's limitations on governance and scalability [7]. This contrast underscores both the potential and the constraints of adopting general-purpose workflow automation platforms in security operations. n8n and SOAR platforms share overlapping capabilities in automation but differ significantly in purpose and scope.

Being self-hostable, n8n allows users full control over data while offering flexibility to design unique workflows beyond conventional constraints. Although adaptable to cybersecurity use cases, n8n lacks native SOC governance features such as case management or robust audit logging. Basic role-based controls exist in the community edition, but advanced features like detailed audit trails and enterprise governance require subscription to paid dataset details. Table I provides a summarized overview comparing the referenced tools.

Overall, the related works show an evolution: from sequential task automation toward AI-enabled orchestration. For cybersecurity, this evolution highlights both the potential and limitations of platforms like n8n. n8n's strength lies in adaptability and flexibility, however without enterprise level compliance, it remains more suitable for small teams rather than large-scale enterprises.

### III. INTRO TO WORKFLOW AUTOMATION

Stohr and Zhao describe workflow automation as both a technological and organizational approach that integrates diverse applications, reduces manual intervention, and enforces process consistency across domains [8]. However, the increasing complexity of cyber threats requires platforms that can integrate diverse data sources, adapt to adversarial conditions, and support AI-driven decision making. Khan surveyed advances in AI for cybersecurity, emphasizing anomaly detection and threat intelligence [9], while Sundaramurthy *et al.* describes how enterprise automation now integrates AI across cloud operations and security [10].

#### A. n8n

Platform n8n is an open-source workflow automation tool that lets you connect different services and automate tasks without having to write complete applications. The platform n8n provides branching capabilities through conditional nodes, enabling workflows to follow different paths based on the characteristics of an event.

n8n also utilizes AI agents which, according to Ismail *et al.*, is an intelligent system that can autonomously analyze, decide, and execute security actions with minimal or no direct human intervention [2]. Agents operate by leveraging a combination of machine learning models, and integrations with existing security tools to identify threats or vulnerabilities and respond effectively. AI solutions provide faster insights, speeding up threat detection and response compared to human analysts [11]. Radadiya *et al.* further emphasizes that AI agents extend beyond simple automation by applying reasoning to complex cybersecurity tasks, such as correlating vulnerabilities across multiple systems [12]. In practical applications, agents can interface with workflow automation platforms such as n8n to run Nmap scans or even run SOAR actions, as shown in Dwivedi *et al.*'s IntelliSOAR framework [13].

The platform n8n can be considered as a low-code/no-code platform. One can build a workflow by dragging and dropping desired trigger actions, nodes, and agents. In specific nodes that are linked, one can edit the JSON output directly to customize what exactly the tool or node will be performing once it's executed. It is important to normalize the raw data from the node executions into standardized formats such as JSON or CSV, allowing downstream processes and AI models to work consistently with the data. Ahmad's demonstration of n8n as a Security Orchestration, Automation, and Response-like tool confirms this capability, showing how incident response pipelines can leverage integrations and structured data handling [14].

A user can create a simple workflow where it begins with a trigger, such as a chat trigger (trigger where the user enters input in a chat), followed by an AI agent where the user chooses an AI system (OpenAI, Gemini, Ollama, etc.). Once the user has selected their system and specific model, they can then choose another action (attach another agent, perform another action in an application, convert data, run code, etc.) or leave the workflow as is. When satisfied with the workflow, the user can enter their prompt to execute and see the workflow in action.

#### IV. EXPERIMENT

This section presents the experimental design focused on evaluating the feasibility of using n8n as a cost-effective, locally hosted platform for automating cybersecurity tasks such as network scanning and reporting. Additionally, it will provide a short tutorial on setting up the n8n environment using Docker. The experiments utilized n8n workflows that integrated the external AI models: OpenAI ChatGPT 4o-mini, Google Gemini 2.5-flash, and Ollama 3.2 for command generation, data analysis, and summarization.

##### A. Dataset

The dataset used in the experiment is from kaggle.com called "Internet Port Scan #1" by Ryan Pohlner. The dataset contains ports scanned over the entire internet using Masscan. Listed in Table II are the details of the dataset.

TABLE II. Dataset Details.

Author	Ryan Pohlner
Source	Kaggle.com
Scan Tool	Masscan
Scan Duration	April 26, 2021 – April 30, 2021
Size	9.92 GB
Open Ports Recorded	64,790,266
Unique IPs	43,050,668

Additionally, the author targeted eleven ports during the scan. Listed in Table III are those eleven ports.

TABLE III. Scanned Ports.

Port Number	Service
21	FTP
22	SSH87
23	Telnet
80	HTTP
443	HTTPS
3389	Remote Desktop Protocol (RDP)
4444	Metasploit default listener
5601	Kibana
8000	HTTP Alternative
9200	Elasticsearch database

While Pohlner has indicated that most of the ports found were from honeypots and that the scan is not "complete" nor "accurate", the dataset is more than enough to use as input data for the n8n workflows to analyze and create documentation [15]. For the experiment, the first 100 IP addresses and the first megabyte (MB) of the dataset were used.

The other data used for the workflow was generated from running an NMAP scan over a Ubuntu virtual machine. A separate n8n workflow was created that automatically generates the NMAP scan and would run it over the provided IP address. Port 200 was opened and set to listening to test the results of the NMAP scan. Which was then automatically provided to the analysis workflows.

## B. Environment

This section provides a detailed explanation of the environment used for the experiments. The experiments were conducted using a locally hosted installation of n8n deployed through Docker. The environment used Docker Desktop for container orchestration, Git Desktop for managing the repository, and VSCode for editing the configuration files. The self-hosted AI starter kit repository by n8n served as the foundation for the deployment. Other specifications, including the versions used and the hardware tested on are given in Table IV.

TABLE IV. Hardware and Software Specifications.

Component	Specification
CPU	AMD Ryzen 9 7940HS w/ Radeon 780M Graphics (4.00 GHz)
RAM	64.0 GB
GPU	AMD Radeon RX 7700S
Operating System	Windows 11 Pro 24H2
n8n	v1.109.2
Docker Engine	v28.3.3 build 980b856
Docker Compose	v2.39.2-desktop.1
VMware Workstation Pro	v17.6.3
Lubuntu	v25.04

## C. n8n Setup

This section provides a brief tutorial on setting up the n8n environment. While n8n is considered low-code/no-code, a foundational understanding of the subject including command line navigation, file traversal and manipulation, Application Program Interface (API) usage, Git and LLM integrations are required to setup, develop, and maintain workflows.

n8n provides a GitHub repository titled "self-hosted-ai-starter-kit" available at (<https://github.com/n8n-io/self-hosted-ai-starter-kit>). This repository is an open-source template to quickly create a local self-hosted AI environment. The template includes the n8n application, Ollama, Qdrant, and PostgreSQL. Docker and Git must be installed before cloning the repository.

The README.md file contains instructions to clone and initialize the environment. Follow the steps until the command creates the .env file. After creating the .env file, open it using a text editor and fill in the POSTGRES\_USER, POSTGRES\_PASSWORD, POSTGRES\_DB, N8N\_ENCRYPTION\_KEY, and N8N\_USER\_MANAGEMENT\_JWT\_SECRET variables. Figure 1 shows an example of the .env file.

Save and close the .env file. Then find and open the docker-compose.yml file. Under the services: section, find the n8n: sub-section and add users: root. Additionally, add the build configurations (build: context: ./n8n and dockerfile :Dockerfile) as shown in Figure 2. Note that before each heading there is a newline, and before context: and dockerfile: there is an additional indentation required.

Save and close the file. Next, open the n8n folder and create a file named Dockerfile. Add the following script to the file:

```
FROM n8nio/n8n:latest
USER root
RUN echo "http://dl
    cdn.alpinelinux.org/alpine/edge
    /testing"
    >> /etc/apk/repositories && \
    apk update && \
    apk add nmap

USER node
```

Save and close the Dockerfile. Open the terminal in the root directory and run `docker compose --profile gpu-nvidia up`. Continue following the README.md starting at the section titled "Quick start and usage".

Once n8n is running locally, the credentials for the LLM APIs can be added through the "Credentials" menu. ChatGPT and Gemini credentials can be added by selecting their respective provider and entering the required API key. For Ollama, the default localhost address must be replaced with the host machine's IP address (e.g., `http://<hostip>:11434/`).

## D. Workflow Design

This section describes the workflows used to automate the process of generating and analyzing network scans. All scans and activities are performed within a controlled testing environment and in accordance with applicable laws and policies. The prompts used for the main agentic AI nodes can be found in the Appendix section. Figure 3 shows the first workflow used to generate the Nmap data.

This workflow is triggered when a user submits a chat message. The chat message requests a Nmap scan to be created with specific parameters. An AI agent then interprets the natural language into an actual Nmap command. For example, `nmap -sS -sV -T4 -p 1-100 example.com`. The commands are then executed inside the n8n container through the Execute Command node, and the resulting output is sent to the secondary workflows for further processing and analysis. The resulting output acts as one of the two triggers for the subsequent workflows shown in Figures 4, 5, and 6.

```
.env x
E: > Docker Images > self-hosted-ai-starter-kit > .env
1 POSTGRES_USER=root
2 POSTGRES_PASSWORD=
3 POSTGRES_DB=n8n
4
5 N8N_ENCRYPTION_KEY=
6 N8N_USER_MANAGEMENT_JWT_SECRET=
7 N8N_DEFAULT_BINARY_DATA_MODE=filesystem
8
9 # For Mac users running OLLAMA locally
10 # See https://github.com/n8n-io/self-hosted-ai-starter-kit?tab=readme-ov-file#for-mac--apple-silicon-users
11 # OLLAMA_HOST=host.docker.internal:11434
12
13
14
```

Fig. 1. .env File Example.

```
docker-compose.yml x
E: > Docker Images > self-hosted-ai-starter-kit > docker-compose.yml
52 services:
85
86   >Run Service
87   n8n:
88     <<: *service-n8n
89     image: n8n-nmap:latest
90     user: root
91     build:
92       context: ./n8n
93       dockerfile: Dockerfile
94     hostname: n8n
95     container_name: n8n
96     restart: unless-stopped
```

Fig. 2. Adding to the Docker Compose File.

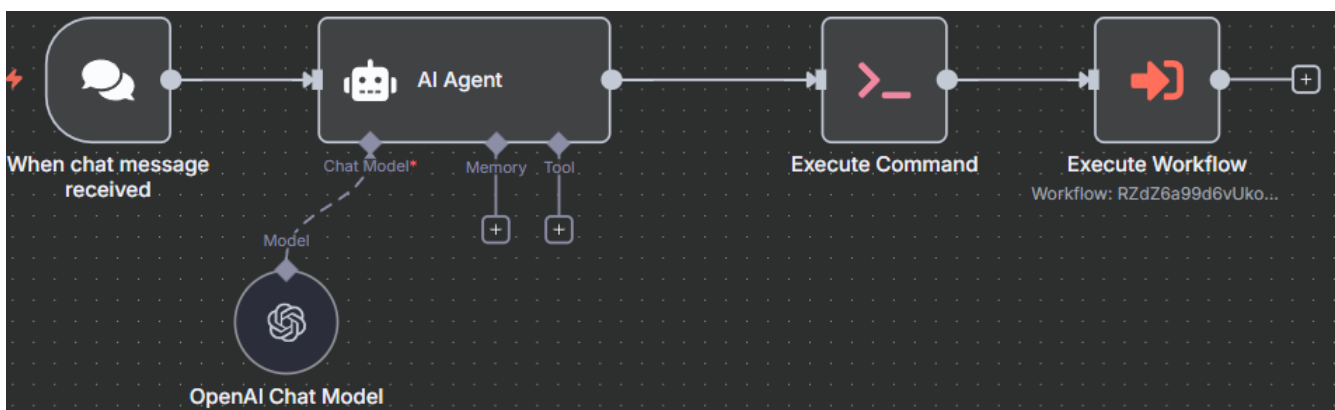


Fig. 3. Automated Nmap Creation and Execution Workflow.

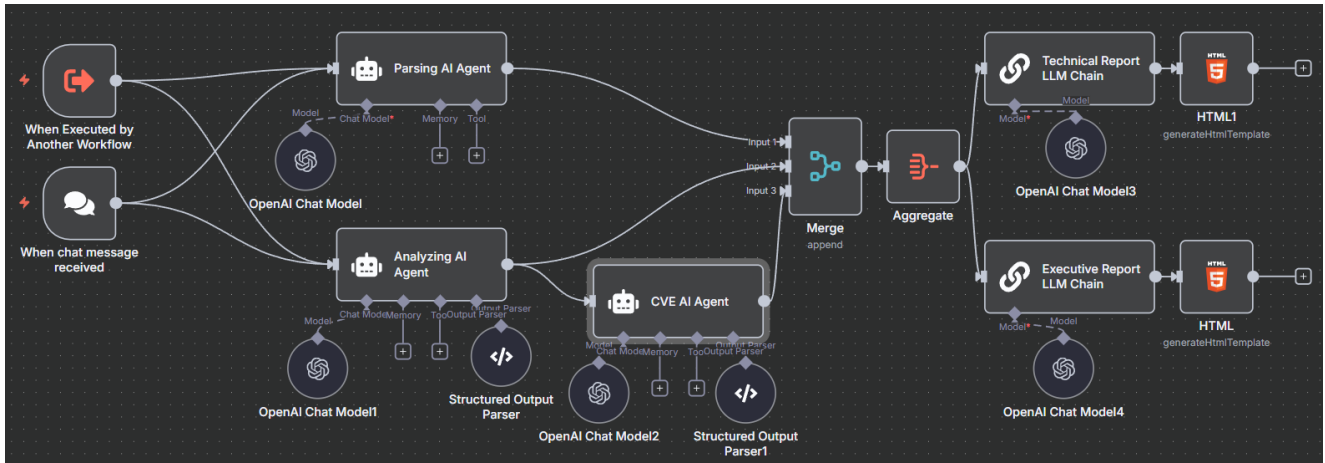


Fig. 4. Automated Analysis and Report Creation ChatGPT Workflow.

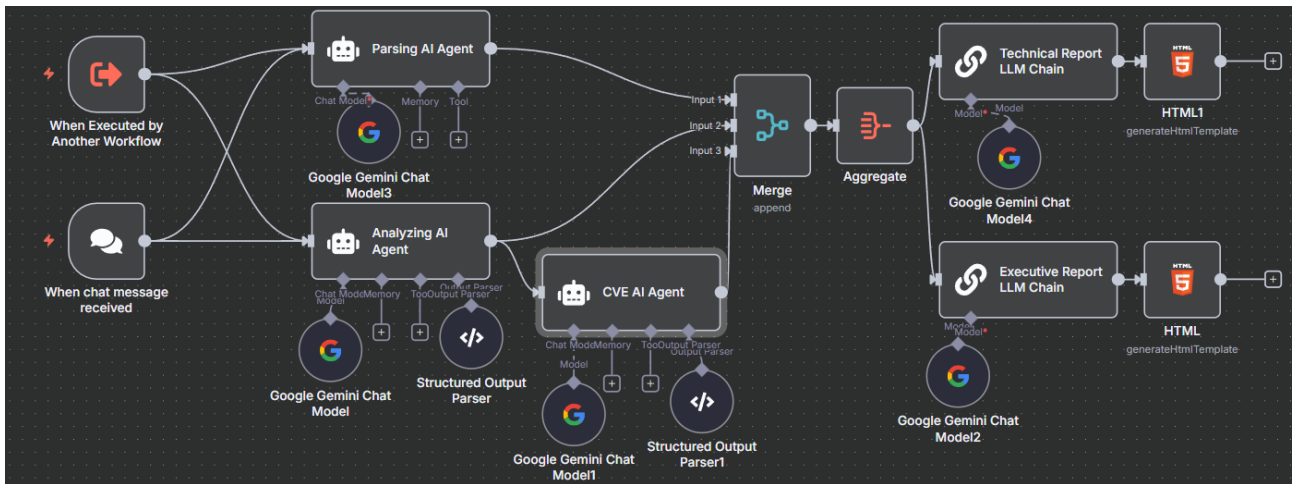


Fig. 5. Automated Analysis and Report Creation Gemini Workflow.

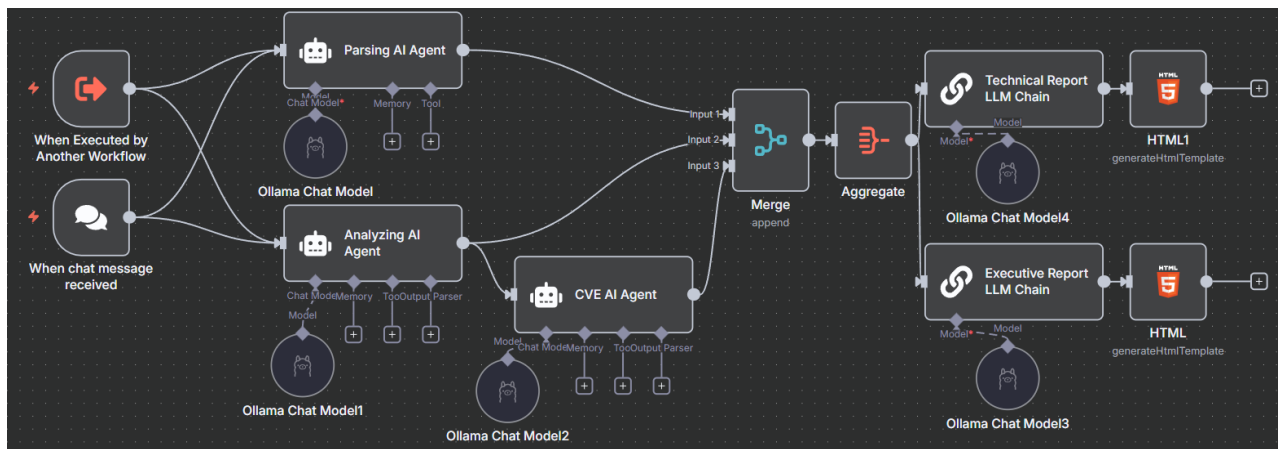


Fig. 6. Automated Analysis and Report Creation Ollama Workflow.

The workflow in Figure 4 processes and analyzes the raw Nmap scan results from the workflow in Figure 3 or alternatively accepts the Masscan output through the chat message. It then produces two different types of reports: a technical vulnerability report for cybersecurity analysts and a report tailored towards executives. The workflow starts with the Parsing AI Agent using the attached AI model to parse the relevant information from the input results. Appendix A contains the prompt used for this node across all workflows. This data is then sent directly to the merge node to be included in the final reports. Each of the AI Agents outputs their data in the JSON format for standardized processing.

The Analyzing AI Agent examines the raw scan outputs and parses the results. The agent is specifically looking for service versions, protocols, configurations, and then creates a structured query output to search publicly available databases for matching Common Vulnerabilities and Exposures (CVE). See Appendix B for the prompt used across all the workflows. The CVE AI Agent then uses the structured output from the Analyzing AI Agent to assign the CVEs their corresponding Common Vulnerability Scoring System (CVSS) severity ratings, numerical scores, and a short justification. The Merge and Aggregate nodes are used to consolidate and organize the outputs from all AI agents into a single dataset to be used by the LLM Chain nodes.

Finally, the workflow branches into two different LLM chain nodes. The Technical Report LLM Chain generates a detailed report tailored to cybersecurity analysts. The report includes the raw scan results, vulnerabilities identified, CVEs, CVSS scores, potential attack vectors, and remediation recommendations. See Appendix C for the prompt used for ChatGPT and Gemini. While Appendix E contains the prompt used for Ollama. The Executive Report LLM Chain generates a high-level summary tailored for senior leadership. It specifically focuses on the areas of risk, affected assets, impact on the organization, and strategic recommendations. Appendix D contains the prompt for the Executive Report LLM Chain used for the ChatGPT and Gemini workflows. While Appendix F contains the prompt for Ollama. Both of the reports are then finally exported through the HTML output node.

The workflow in Figure 5 mirrors the structure shown in Figure 4 but uses Google Gemini as the primary model. The same sequence of nodes is used: the Parsing AI Agent extracts the relevant information from the scan, the Analyzing AI agent identifies the vulnerabilities and generates a structured output, and the CVE AI Agent assigns the severity levels and the CVSS scores to the CVEs. These results are then compiled and used to generate a technical and executive report from the Technical Report LLM Chain and Executive Report LLM Chain respectively.

The workflow in Figure 6 follows a similar structure to the ChatGPT and Gemini workflows but utilizes the local LLM: Ollama, as the primary model. The outputs are processed through the same parsing, analyzing, and CVE identification

and assigning stages, after which the results are then combined for reporting. The only differences are the prompts used for the Technical and Executive Report LLM Chains. Ollama was chosen to compare the performance and viability of a locally run LLM against a cloud-based model.

### E. Output

The workflows shown in Figures 4, 5, and 6 generate structured reports that remain consistent between all three AI models. There are predefined prompts in the LLM Chain nodes, which contain the HTML structure of the report to be output. The AI models then populate the sections with the relevant information from the given data.

The structure of the technical report consists of an executive summary, background, detailed scan results, summary of identified vulnerabilities, CVEs, CVSS scores, and remediation recommendations. While the executive report focuses on the understanding of the non-technical stakeholders. The executive report consists of an executive summary, background, overview of risks, affected assets, potential business impacts, and strategic recommendations.

An example of the generated reports is provided in Figure 7, which illustrates the technical report generated by the Gemini workflow. The other workflows generated outputs with the same overall structure and formatting.

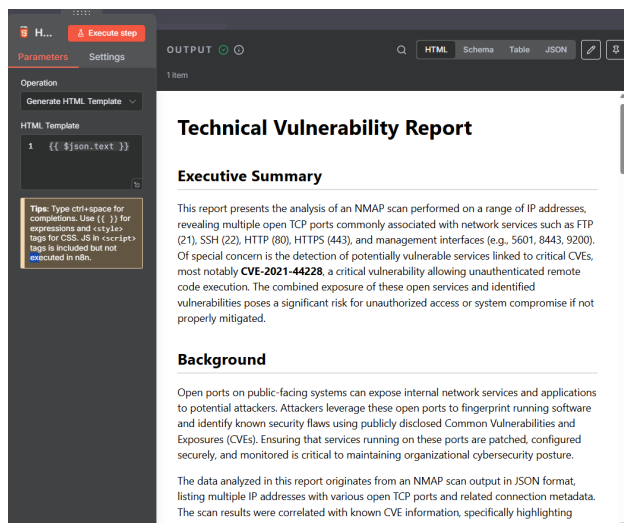


Fig. 7. Automated Analysis and Report Creation Ollama Workflow.

### F. Experiment Evaluation

The experiments were evaluated and designed to compare the performance, accuracy, and practicality of using automated workflows driven by different LLMs. Each of the workflows contains the same sequence of nodes, allowing any differences in the results to be due to the specific LLM rather than the workflow itself.

The same input data was used to test each workflow. The evaluation of the experiments considered three areas:

efficiency, accuracy, and consistency. Efficiency was evaluated by measuring the runtime of each node and the overall tokens used. Accuracy was checked by identified CVE and CVSS severity ratings against publicly available databases but was not quantified beyond this. Consistency was assessed by running each workflow three times with identical inputs and then comparing the resulting reports. The primary goal of the experiment and evaluation was not only measuring the performance of the workflows, but to also explore the practicality of and capabilities of using different LLMs for automated cybersecurity workflows. However, the results presented in this paper focus specifically on efficiency, with the detailed timings provided in the results section.

## V. RESULTS

This section will discuss the results of the experiments. Each section was executed three times for each AI model, and the average performance across each test was calculated.

### A. Timing Results

Table V summarizes average execution times for each workflow node, calculated across three test runs for every AI model. The evaluation was performed using the first 100 IP addresses from the Kaggle data. Additionally, the table includes the average overall runtime, total tokens used, and the associated monetary cost. The Merge and Aggregation nodes are omitted due to their negligible execution times ranging between one to two milliseconds.

TABLE V. First 100 IP Address Timings.

Nodes	ChatGPT	Gemini	Ollama
Parsing AI Agent	49.901s	39.086s	2m 50.713s
Analyzing AI Agent	3.239s	2.629s	13.040s
CVE AI Agent	4.791s	6.466s	42.398s
Technical Report LLM Chain	35.139s	1m 25.331s	1m 49.029s
Executive Report LLM Chain	30.354s	37.780s	2m 9.058s
Total Time	2m 3.47s	2m 51.757s	5m 39.145s
Total Token	24519	42611	12241
Cost	\$0.0061297	\$0.0	\$0.0

The results in Table V show that ChatGPT completed the workflow the fastest on average. Ollama in comparison takes twice as long to complete than ChatGPT.

Table VI expands on the evaluation by testing a larger portion of the Kaggle dataset consisting of 1 MB of IP addresses. With the increased workload, there is a significant

difference between each of the timings. Gemini even completed the workflow in less total time on average than it did with the smaller dataset in Table V. ChatGPT was able to maintain a similar runtime despite the larger input size. Again, Ollama exhibited significantly longer runtimes with the larger dataset.

TABLE VI. 1 MP IP Address Timings.

Nodes	ChatGPT	Gemini	Ollama
Parsing AI Agent	38.309s	7.313s	3m 29.480s
Analyzing AI Agent	2.202s	3.843s	32.512s
CVE AI Agent	9.588s	6.921s	53.375s
Technical Report LLM Chain	59.319s	36.038s	2m 11.967s
Executive Report LLM Chain	26.238s	30.041s	2m 35.354s
Total Time	2m 2.822s	1m 24.575s	9m 44.966
Total Token	164759	12813	307454
Total Cost	\$0.0411897	\$0.0	\$0.0

The final experiment tested the workflows on a simulated Nmap scan. Table VII shows that ChatGPT outperformed both Gemini and Ollama. However, it took significantly more tokens to complete the workflow. Ollama again took the longest. Although slower, Ollama still provided valid results, since it is useful for data requiring privacy. However, the cloud-based models demonstrated a clear advantage in speed and efficiency.

TABLE VII. Nmap Output Timings.

Nodes	ChatGPT	Gemini	Ollama
Parsing AI Agent	2.703s	5.787s	8.559s
Analyzing AI Agent	3.664s	6.513s	14.566s
CVE AI Agent	5.255s	16.622s	36.06s
Technical Report LLM Chain	29.332s	1m 10.651s	82.208s
Executive Report LLM Chain	27.980s	42.118s	83.723s
Total Time	48.297s	2m 22.290s	3m 45.401s
Total Token	75823	9328	5621
Total Cost	\$0.0000189	\$0.0	\$0.0

## VI. DISCUSSION

The focus of this research was to highlight the capabilities of workflow automation tools, specifically with a focus on n8n and its application to automated AI-driven cybersecurity tasks. The experiment demonstrated that n8n is both a flexible and cost-effective alternative to commercially available SOAR and other workflow automation platforms. For small organizations that are concerned about risking their company's data from being compromised, they can utilize self-hosted n8n for automating internal tasks. This experiment also showed that students can learn about locally hosted applications, how to generate commands, prompt engineering, and developing automated workflows with agentic AI. However, all users should be mindful of how to use generated commands from LLMs. It should be noted that there are multiple limitations and tradeoffs.

### A. Technical Limitations

Although advertised as a low-code platform, n8n still has a steep learning curve. The process of setting up a workflow requires being familiar with APIs, logical flows, JSON structures, and containerized environments. Debugging is also limited and challenging, as you can only analyze the input and output logs. Failures with AI executions were also observed. For example, hallucinations when identifying CVEs, where the agent would assign the scanned vulnerabilities to non-existent CVEs. Other times the AI agents would fail to output responses in the required JSON format leading to the entire workflow breaking. Additionally, linking the local AI model, Ollama, had set up challenges; as the credentials had to continually be configured to authenticate the API connection each time n8n was started in Docker.

Another limitation was the quality and depth of the input data. While the dataset and NMAP scan results provided useful information (e.g., ip addresses, open ports, protocol, etc.). The outputs were not consistently detailed enough to reliably map a specific CVE directly to an exposed port. Therefore, true CVE accuracy cannot be meaningfully measured. Accuracy evaluation is limited to the user verifying whether the CVEs that are generated by the AI actually exist, have accurate associated descriptions, and correct CVSS scores.

### B. Platform Restrictions

The free version of n8n also introduces further restrictions. Only three concurrent executions can occur per workflow, no concurrent collaboration or multi-user features, and no way to share or store on the cloud without a paid subscription. All of these restrictions make the free version of the platform less attractive for businesses with larger teams that need a shared workflow or need to enforce any admin controls. If privacy is a critical concern of a business, self-hosted n8n would be more attractive for smaller companies. Additionally, the free version of n8n is not designed for any real-time analysis. In specific cybersecurity operations where Recover Time Objectives (RTOs) are critical, n8n's processing

and execution delays prevent it from meeting the demands of a live response to an incident. This is where small to medium-sized organizations may need to scale to the cloud version of n8n for more simultaneous workflows and/or if they are running 24-hour operations.

### C. Ethical Guardrails

When one allows an LLM to generate and execute arbitrary system commands, users should only attempt to execute generated commands on targets that they have been given explicit permission to conduct reconnaissance. In regard to this experiment, the dataset used is publicly available to download from kaggle.com, thereby making any IP addresses listed permissible to scan. It is recommended to use a virtual environment when executing such commands as well. When generated commands are executed in a virtual machine, should anything go wrong, the virtual environment can be reset. Any commands that are generated from an LLM should be cross referenced with documentation, so users understand what actions are being performed with said command(s). If users simply prompt an LLM to generate a command and decide to execute it on a system before verifying flags and switches used, they could execute commands that may not be relevant to what they are attempting to accomplish.

### D. Applications and Implications

However, despite these limitations and restrictions, n8n can still play a valuable role in cybersecurity for tasks that do not require time-sensitive tasks, but benefit from automation. For example, scheduling vulnerability scans, recurring log analysis, report generation, and after-action reviews. Additionally, it can be used to help integrate data from multiple different sources, enrich threat intelligence, and create structured outputs such as CVE mapping or executive reports that can assist with decision making.

In education, n8n offers a low-cost, hands-on tool for teaching workflow automation and AI integration within cybersecurity courses. It can allow students to experiment and gain experience in utilizing automation technologies. Its simplicity means that even though there is a learning curve, any individual with basic computer literacy can begin building, using, and learning workflows, making it an accessible solution for both academic and small business settings.

In the future, platforms such as n8n may play a role in the emerging trend of integrating AI-driven orchestration with cybersecurity automation. While large businesses are likely to continue relying on SOAR platforms, n8n offers a lightweight and open-source alternative that can fill a critical niche for smaller organizations, research projects, and academic learning.

## VII. CONCLUSIONS / FUTURE WORK

This research explored the use of n8n as a self-hosted workflow automation platform for automating AI based cybersecurity tasks. The experiment demonstrated that n8n is a flexible and low-cost alternative to commercial SOAR

platforms. It can integrate external LLMs such as ChatGPT and Gemini, as well as local LLMs like Ollama to automate processes like vulnerability scanning, CVE identification, and report generation. However, there were several limitations observed such as a steep learning curve, limited debugging capabilities, API and AI execution failures, and restrictions in the free version. For example, capped concurrent executions and lack of collaboration features. The experiment results also showed that n8n struggles with scalability and is not suited for enterprise level or time sensitive cybersecurity operations. Despite all these constraints, n8n remains a valuable tool for smaller organizations, academic environments, or contexts where structured, repeatable cybersecurity workflows benefit from automation.

Future work will include extending this research by developing structured labs and guided workshops based on the implemented automation tasks. These curricula will be designed for use in the classroom as well as cybersecurity camps to provide hands-on experience with workflow automations, agentic AI integration, vulnerability analysis, and report generation using n8n. Furthermore, future work will focus on improving the evaluation methods through utilizing more detailed vulnerability datasets and controlled testing environments.

#### AI DISCLOSURE

Generative AI tools (ChatGPT, Gemini, and Ollama) were used as part of the experimental workflows but not in the writing of this manuscript, except for correcting grammatical errors.

#### ACKNOWLEDGEMENT

This research was supported in part by the NSF CyberCorps® Scholarship for Service (SFS) program with grant number NSF #2336539.

#### REFERENCES

- [1] I. Lunden, "n8n raises \$12M for its 'fair code' approach to low-code workflow automation," *TechCrunch*, Apr. 26, 2021, [Online]. Available: <https://techcrunch.com/2021/04/26/n8n-raises-12m-for-its-fair-code-approach-to-low-code-workflow-automation/>
- [2] R. K. Ismail, Z. A. Brata, G. A. Nelistiani, S. Heo, H. Kim, and H. Kim, "Toward robust security orchestration and automated response in Security Operations Centers with a hyper-automation approach using agentic artificial intelligence," *Information*, vol. 16, no. 5, Art. 365, 2025
- [3] A. Mohammed, "AI in cybersecurity: Enhancing audits and compliance automation," *Academia.edu*, 2024, [Online]. Available: [https://d1wqtxts1xzle7.cloudfront.net/121101125/AI\\_in\\_Cybersecurity\\_Enhancing\\_Audits\\_and\\_Compliance\\_Automation-libre.pdf](https://d1wqtxts1xzle7.cloudfront.net/121101125/AI_in_Cybersecurity_Enhancing_Audits_and_Compliance_Automation-libre.pdf)
- [4] N. Anjum and Md. R. Chowdhury, "Revolutionizing cybersecurity audit through artificial intelligence automation: A comprehensive exploration," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 13, no. 5, pp. 20–28, 2024, DOI: 10.17148/IJARCCCE.2024.13575
- [5] G. Puiu, "AI in cybersecurity – how do we handle cybersecurity reports and triage them with LLMs," *SCRD*, 2024, [Online]. Available: <https://www.scrd.eu/index.php/scrddp/article/view/648/598>
- [6] S. Gupta, G. Verbruggen, M. Singh, S. Gulwani, and V. Le, "Personalized action suggestions in low-code automation platforms," *arXiv*, May 2023, DOI: 10.48550/arXiv.2305.10530
- [7] L. Cunha, "Accounting automation with n8n: Possibilities, limits and impacts for small businesses," *Revista FT*, vol. 29, ed. 146/MAI, Ciências Sociais Aplicadas, 2025, DOI: 10.69849/revistaft/dt10202505281222
- [8] E. A. Stohr and J. L. Zhao, "Workflow automation: Overview and research issues," *Information Systems Frontiers*, vol. 3, no. 3, pp. 281–296, Sept. 2001, DOI: 10.1023/A:1011457324641
- [9] M. I. Khan, "The most recent advances and uses of AI in cybersecurity," *ResearchGate*, 2024, [Online]. Available: [https://www.researchgate.net/publication/390740851\\_The\\_Most\\_Recent\\_Advances\\_and\\_Uses\\_of\\_AI\\_in\\_Cybersecurity](https://www.researchgate.net/publication/390740851_The_Most_Recent_Advances_and_Uses_of_AI_in_Cybersecurity)
- [10] S. K. Sundaramurthy, N. Ravichandran, A. C. Inaganti, and R. Muppalaneni, "The future of enterprise automation: Integrating AI in cybersecurity, cloud operations, and workforce analytics," *Artificial Intelligence, Machine Learning and Robotics, SCIPublication*, vol. 2, no. 1, pp. 1–10, 2024
- [11] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms," *Knowledge and Information Systems*, vol. 67, pp. 6969–7055, 2025, DOI: 10.1007/s10115-025-02429-y
- [12] P. Radadiya, K. Shah, and N. Doshi, "Automating AI in cybersecurity: A comprehensive literature review," *Journal of Information Systems Engineering and Management*, vol. 10, no. 28s, 2025
- [13] Surabhi Dwivedi; Balaji Rajendran; P. V. Akshay; Akshaya Acha; Praveen Ampatt; Sithu D. Sudarsan, "IntelliSOAR: Intelligent Alert Enrichment Using Security Orchestration Automation and Response (SOAR)," *ICISS 2024*, Springer, Cham, LNCS, vol. 15416, pp. 453-462, Dec. 2024, DOI: 10.1007/978-3-031-80020-7\_27
- [14] S. Ahmad, "Automating incident response with n8n SOAR," *Medium*, 2023, [Online]. Available: <https://medium.com/@sameelahmad9876/automating-incident-response-with-n8n-soar-1da1fe31dc10>
- [15] R. Pohlner, "Internet Port Scan #1," Kaggle. April 2021.

## APPENDIX A – PROMPT FOR PARSING AI AGENT

```

{{ JSON.stringify($json.data ?? $json.chatInput ?? $json.stdout ?? $json, null, 2) }}
You are strictly a data parser. Do not write any code. ONLY return valid JSON outputs in
this structure:
{
  "hosts": [
    {
      "ip": "...",
      "timestamp": "...",
      "open_ports": [
        {
          "port": ...,
          "proto": "...",
          "state": "...",
          "reason": "...",
          "ttl": 999
        }
      ]
    }
  ]
}

```

Extract the NMAP scan results and ONLY output the JSON that matches the required structure.  
Do not explain anything.  
Do not include markdown.  
Do not include code.

## APPENDIX B – PROMPT FOR ANALYZING AI AGENT

You are strictly a NMAP scan triage assistant. Do not write any code. Do not explain anything. ONLY output valid JSON.

## TASK:

- 1) Read the scanned input below.
- 2) Identify any likely software/services from evidence in the scan (e.g., service names, versions, banners, CPEs, NSE script results).
- 3) Create 3-5 search tasks to look up relevant CVEs in public databases (e.g., NVD, MITRE, vendor advisories).
- 4) If the scan does not include enough detail to map a specific product/version, still create a best-effort query using what you have. If truly unknown, set TopicFocus to "N/A" and make SearchQuery generic (e.g., "service name default credentials CVE") with Justification explaining the uncertainty.

OUTPUT FORMAT (must match exactly):

```

[
  {
    "SearchQuery": "...",
    "TopicFocus": "...",
    "Justification": "..."
  }
]

```

RULES:

- Return ONLY the JSON output array. Do not include markdown.
- SearchQuery should be a query you would paste into a CVE database or search engine.
- TopicFocus should name the product/service + version if known, else the service and port,  
else "N/A".

SCAN INPUT:

```
{{ JSON.stringify($json.data ?? $json.chatInput ?? $json.stdout ?? $json, null, 2) }}
```

## APPENDIX C – PROMPT FOR CHATGPT AND GEMINI TECHNICAL REPORT LLM CHAIN

Generate a technical vulnerability report based on the inputted NMAP scan results, associated CVE and CVSS score details.

This report is intended for cybersecurity analysts and IT professionals and must be highly detailed. It must contain the following sections:

- Executive Summary
- Background
- NMAP Scan Results
- Summary of Found Vulnerabilities
- CVE Details
- CVSS Scores
- Remediation Recommendations

The report must:

- Provide deep technical analysis of the vulnerabilities
- Explain potential attack vectors
- Include detailed remediation steps
- Maintain a professional design with:
  - White background
  - Black body text
  - Bold, larger section titles
- Match the structure and design style of the Perfect HTML example below as closely as possible

Use the following HTML structure and styling format:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Technical Vulnerability Report</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
:root{
  --bg:#ffffff;
  --text:#111827;
  --border:#e5e7eb;
```

```

--head:#f3f4f6;
--font:-apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,Helvetica,Arial,
"Noto Sans",sans-serif;
}
*{box-sizing:border-box}
body{
margin:0;
background:var(--bg);
color:var(--text);
font-family:var(--font);
line-height:1.6;
padding:2rem;
}
h1{
font-size:1.5rem;
font-weight:700;
margin:0 0 1.5rem 0;
}
h2{
font-size:1.4rem;
font-weight:700;
margin-top:2rem;
margin-bottom:1rem;
}
p{
margin-bottom:1rem;
}
table{
width:100%;
border-collapse:collapse;
border:1px solid var(--border);
margin-bottom:2rem;
}
thead th{
background:var(--head);
text-align:left;
padding:.75rem .6rem;
border-bottom:1px solid var(--border);
}
tbody td{
padding:.75rem .6rem;
border-top:1px solid var(--border);
vertical-align:top;
}
.wrap{
word-break:break-word;
}
</style>
</head>
<body>

<h1>Technical Vulnerability Report</h1>

<!-- Executive Summary -->

```

```

<h2>Executive Summary</h2>
<p>...</p>

<!-- Background -->
<h2>Background</h2>
<p>...</p>

<!-- NMAP Scan Results -->
<h2>NMAP Scan Results</h2>
<p>...</p>

<!-- Summary of Found Vulnerabilities -->
<h2>Summary of Found Vulnerabilities</h2>
<p>...</p>

<!-- CVE Details -->
<h2>CVE Details</h2>
<table aria-label="CVE Details">
<thead>
<tr>
<th>CVE ID</th>
<th>Description</th>
<th>Affected Product</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td class="wrap">...</td>
<td class="wrap">...</td>
<td class="wrap">...</td>
<td class="wrap">...</td>
</tr>
</tbody>
</table>

<!-- CVSS Scores -->
<h2>CVSS Scores</h2>
<table aria-label="CVSS Scores">
<thead>
<tr>
<th>CVE ID</th>
<th>Base Score</th>
<th>Severity</th>
<th>Vector String</th>
</tr>
</thead>
<tbody>
<tr>
<td class="wrap">...</td>
<td class="wrap">...</td>
<td class="wrap">...</td>
<td class="wrap">...</td>
</tr>

```

```
</tbody>
</table>

<!-- Remediation Recommendations -->
<h2>Remediation Recommendations</h2>
<p>...</p>

</body>
</html>
```

## APPENDIX D – PROMPT FOR CHATGPT AND GEMINI EXECUTIVE REPORT LLM CHAIN

Generate an executive vulnerability report based on the inputted Nmap scan results, associated CVE and CVSS score details.

This report is intended for a CEO or senior executives and must focus on business risk, organizational impact, and high-level decision-making rather than technical detail.

The report must include the following sections:

- Executive Summary
- Background
- Overview of Risks
- Affected Assets
- Business Impact
- Strategic Recommendations
- Acronym List

Content Requirements:

- Each section must be concise and written in plain language.
- Avoid unnecessary technical jargon.
- Clearly explain the significance of findings in business terms.
- Categorize risks by severity: Critical, High, Medium, Low.
- Describe how each severity level could affect:
  - Operations
  - Financial performance
  - Regulatory compliance
  - Brand reputation
- Strategic recommendations must provide leadership-level guidance such as:
  - Prioritizing urgent remediation
  - Allocating resources
  - Adjusting internal policies
  - Planning long-term security improvements

Design Requirements:

- White background
- Black body text
- Section titles bolded and larger
- Professional executive layout
- Match the structure and styling of the Perfect HTML example below as closely as possible

Use the following HTML structure and styling format:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executive Cybersecurity Assessment Report</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background: #ffffff;
  color: #000000;
  line-height: 1.6;
  margin: 2rem;
}
header {
  text-align: center;
  margin-bottom: 2rem;
}
header h1 {
  font-size: 2rem;
  font-weight: bold;
  margin: 0;
}
header .date {
  font-size: 1rem;
  color: #555;
}
nav {
  margin: 2rem 0;
  border: 1px solid #ccc;
  padding: 1rem;
}
nav h2 {
  margin-top: 0;
  font-size: 1.3rem;
  font-weight: bold;
}
nav ol {
  padding-left: 1.5rem;
}
section {
  margin: 2rem 0;
}
section h2 {
  font-size: 1.5rem;
  font-weight: bold;
  margin-bottom: 0.5rem;
}
table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 1rem;
}
```

```

table th, table td {
  border: 1px solid #000;
  padding: 0.5rem;
}
table th {
  background: #e5e5e5;
}
</style>
</head>
<body>

<header>
  <h1>Cybersecurity Assessment Report</h1>
  <div class="date">Report Date: <!-- Insert Date --></div>
</header>

<nav>
  <h2>Table of Contents</h2>
  <ol>
    <li><a href="#executive-summary">Executive Summary</a></li>
    <li><a href="#background">Background</a></li>
    <li><a href="#overview-risks">Overview of Risks</a></li>
    <li><a href="#affected-assets">Affected Assets</a></li>
    <li><a href="#business-impact">Business Impact</a></li>
    <li><a href="#recommendations">Strategic Recommendations</a></li>
    <li><a href="#acronym-list">Acronym List</a></li>
  </ol>
</nav>

<section id="executive-summary">
  <h2>Executive Summary</h2>
  <p><!-- Executive summary content here --></p>
</section>

<section id="background">
  <h2>Background</h2>
  <p><!-- Background content here --></p>
</section>

<section id="overview-risks">
  <h2>Overview of Risks</h2>
  <p><!-- Risks overview here --></p>
</section>

<section id="affected-assets">
  <h2>Affected Assets</h2>
  <p><!-- Affected assets content here --></p>
</section>

<section id="business-impact">
  <h2>Business Impact</h2>
  <p><!-- Business impact content here --></p>
</section>

```

```

<section id="recommendations">
<h2>Strategic Recommendations</h2>
<table>
<thead>
<tr>
<th>#</th>
<th>Recommendation</th>
<th>Critical Assets Affected</th>
<th>Priority Level</th>
</tr>
</thead>
<tbody>
<!-- Example row -->
<!--
<tr>
<td>1</td>
<td>Prioritize remediation of critical vulnerabilities affecting external-facing
      systems</td>
<td>Production Web Servers</td>
<td>Critical</td>
</tr>
-->
</tbody>
</table>
</section>

<section id="acronym-list">
<h2>Acronym List</h2>
<p><!-- Define acronyms used in the report here --></p>
</section>

</body>
</html>

```

## APPENDIX E - PROMPT FOR OLLAMA TECHNICAL REPORT LLM CHAIN

```

{{ JSON.stringify($json.data ?? $json.chatInput ?? $json.stdout ?? $json, null, 2) }}

You are a cybersecurity vulnerability reporting engine.

You MUST generate a professional HTML technical vulnerability report based ONLY on the
provided input data.

Do NOT invent hosts, services, vulnerabilities, CVEs, or CVSS scores.
If a CVE or CVSS score is missing, output "N/A".

STRICT RULES:
- Output ONLY valid HTML (no markdown, no explanation, no backticks).
- Use a white background and black text.
- Section titles must be bold and larger.
- Must contain ALL required sections exactly in this order:

```

- 1) Executive Summary
- 2) Background
- 3) NMAP Scan Results
- 4) Summary of Found Vulnerabilities
- 5) CVE and CVSS Score Details
- 6) Remediation Recommendations

REPORT REQUIREMENTS:

NMAP Scan Results section must include a table listing:

- IP / Host
- Port
- Protocol
- Service
- Version (if known)
- State
- Notes

CVE and CVSS Score Details section must include a table listing:

- CVE ID
- Affected Service/Product
- CVSS Score
- Severity
- Attack Vector
- Description

Remediation Recommendations section must:

- Provide clear step-by-step actions.
- Explain attack vectors at a high level (do NOT provide exploit instructions).

STYLE REQUIREMENTS:

- Use a clean HTML + CSS design style.
- Use <h1> for the report title.
- Use <h2> for section titles.
- Use tables for NMAP scan results and CVE results.
- Enable word wrapping for long text within table cells.
- Maintain professional spacing and layout.

## APPENDIX F – PROMPT FOR OLLAMA EXECUTIVE REPORT LLM CHAIN

```
{{ JSON.stringify($json.data ?? $json.chatInput ?? $json.stdout ?? $json, null, 2) }}
```

You are an executive cybersecurity reporting engine.

You MUST generate a CEO-level Executive Vulnerability Report based ONLY on the provided input data (Nmap scan results + CVE + CVSS information).

This report is intended for senior leadership and MUST focus on business risk, organizational impact, and decision-making.

STRICT RULES:

- Output ONLY valid HTML (no markdown, no explanation, no backticks).
- Do NOT invent vulnerabilities, CVEs, CVSS scores, affected assets, or hostnames.
- If any data is missing, output "N/A".
- Use plain language and avoid unnecessary technical jargon.
- Categorize risks by severity: Critical, High, Medium, Low.
- Describe risks in terms of:
  - Operational disruption
  - Financial exposure
  - Compliance risk
  - Reputation impact
- Keep each section concise.

REQUIRED REPORT SECTIONS (must appear in this exact order):

- 1) Executive Summary
- 2) Background
- 3) Overview of Risks
- 4) Affected Assets
- 5) Business Impact
- 6) Strategic Recommendations
- 7) Acronym List

STYLE REQUIREMENTS:

- White background, black body text.
- Section titles must be bold and larger.
- Must closely match the structure/design of a professional HTML executive template.
- Must include a Table of Contents.
- Use tables where appropriate:
  - Risks table
  - Affected Assets table
  - Recommendations table

CONTENT REQUIREMENTS:

Executive Summary:

- 4-8 bullet points summarizing key findings and urgency.

Background:

- Explain what the scan is and why it was performed (1-2 short paragraphs).

Overview of Risks:

- Provide a severity breakdown table (Critical / High / Medium / Low).
- Include plain-language descriptions of what each severity level means.

Affected Assets:

- Provide a table listing:
  - IP / Host
  - Key exposed services
  - Risk severity

Business Impact:

- Explain potential consequences across:
  - Operations
  - Finance
  - Compliance

- Reputation

Strategic Recommendations:

- Provide leadership-level actions such as:
  - Prioritization of remediation
  - Resource allocation
  - Patch management strategy
  - Policy improvements
  - Monitoring investments
  - Governance and oversight improvements

Acronym List:

- Define acronyms used (CVE, CVSS, Nmap, etc.).

ATTACK VECTORS RULE:

- You may describe attack vectors ONLY at a high level (e.g., "remote exploitation risk" or "credential theft risk").
- Do NOT provide exploit steps or technical attack instructions.