Structured Extraction of Vulnerabilities in OpenVAS and Tenable WAS Reports Using LLMs

Beatriz Machado, Douglas Lautert, Cristhian Kapelinski, Diego Kreutz[1]

[1]AI Horizon Labs – PPGES – Federal University of Pampa (UNIPAMPA)

{beatrizmachado,douglaslautert,cristhianavilla}.aluno@unipampa.edu.br
diegokreutz@unipampa.edu.br

Abstract. *This paper proposes an automated LLM-based method to extract and structure vulnerabilities from OpenVAS and Tenable WAS scanner reports, converting unstructured data into a standardized format for risk management. In an evaluation using a report with 34 vulnerabilities, GPT-4.1 and DeepSeek achieved the highest similarity to the baseline (ROUGE-L greater than 0.7). The method demonstrates feasibility in transforming complex reports into usable datasets, enabling effective prioritization and future anonymization of sensitive data.*

1. Introduction

Vulnerability scanners such as OpenVAS and Tenable WAS are widely used to identify flaws in web applications; however, they produce structurally heterogeneous reports, which complicates automated analysis and integration with machine learning models. This inconsistency, combined with the large volume of reported vulnerabilities, intensifies the challenge of prioritization, especially in institutions with limited resources. Data from SGIS indicate that more than 500,000 vulnerabilities remained unaddressed in 2017 [Rede Nacional de Ensino e Pesquisa 2018], a trend that increases with the expansion of the attack surface and the continuous growth of threats, as highlighted by the RNP Security Report [Rede Nacional de Ensino e Pesquisa 2024].

In this context, this work proposes a method based on Large Language Models (LLMs) to extract vulnerabilities from OpenVAS and Tenable WAS reports. The developed tool, called *Vulnerability Extractor*[1], converts these reports into a structured format to enable subsequent analysis, prioritization, and the construction of datasets usable by machine learning models. The approach aims to standardize and harmonize the extracted information, ensuring consistency across different tools and significantly reducing manual processing overhead. Additionally, the method is designed for future integration with anonymization modules, enabling the creation of secure and shareable datasets among institutions.

2. Vulnerability Reports

OpenVAS and Tenable WAS are among the leading tools for automatic vulnerability detection, but they present structural and semantic differences that directly impact the extraction and analysis of results. Understanding these distinctions is essential for ensuring greater consistency in information extraction and data standardization from both sources.

[1]https://github.com/AnonShield/Vulnerability_Extractor

Table 1. Fields present in OpenVAS reports.

Element	Description
Summary	Brief summary of the identified issue.
Vulnerability Detection Result	Indicates vulnerable URLs, ports, or services detected by the scanner.
Impact	Explains the potential consequences of exploiting the vulnerability.
Solution	Provides recommendations to mitigate or remediate the issue.
Affected Software/OS	Identifies the affected software, operating system, or component.
Vulnerability Insight	Details the origin and exploitation mechanism of the vulnerability.
Vulnerability Detection Method	Describes the method or script used by the scanner to identify the flaw.
Log Method	Specifies techniques or logging approaches used during detection.
References	Lists CVE identifiers, links, or other relevant references.

OpenVAS, as shown in Table 1, prioritizes a high level of technical detail, including fields such as *Vulnerability Insight*, which describe the nature and causes of vulnerabilities [Greenbone 2025]. On the other hand, Tenable WAS (see Table 2) adopts a more risk-management-oriented approach, incorporating sections such as *Risk Information*, which directly support remediation prioritization [Tenable 2025].

Table 2. Fields present in Tenable WAS reports.

Element	Description
Affected Application	Information about the affected application, including name, first and last detection dates.
Description	Details of the Tenable plugin responsible for identifying the vulnerability and contextualizing its behavior.
Solution	Recommended mitigation actions, including official fixes when available.
See Also	External links and references that expand the technical description.
Vulnerability Properties	General properties such as severity, exploitability, publication date, and remediation status.
Discovery	Records the first and last detection dates, providing monitoring history.
VPR Key Drivers	Factors used to calculate the *Vulnerability Priority Rating* (VPR), such as impact and exploitation trends.
Plugin Details	Technical information about the detection plugin, including version and dependencies.
Risk Information	Associated risk metrics (CVSSv2, CVSSv3, CVSSv4), attack vectors, and risk modifications such as *Accept* or *Recast*.
Reference Information	External references related to the vulnerability, exploit, patch, or security bulletins.

Although some scanners, such as OpenVAS, provide export methods in structured formats like XML and CSV [Greenbone 2025], the structural and semantic heterogeneity across different tools remains a significant challenge, as each scanner has proprietary incompatible schemas where some fields may not exist in both tools or are labeled differently despite containing similar information.

Table 3 illustrates how the same vulnerability may be represented differently across vulnerability scanners, reflecting variations in structure and detail. While Open-

VAS tends to use a more granular and technically oriented organization, Tenable WAS emphasizes consolidated information with a focus on risk management and prioritization. This structural heterogeneity not only complicates automated analysis but also required adaptations in the extraction process to ensure correct interpretation and standardization of data.

Table 3. Comparison of OptionsBleed in OpenVAS and Tenable WAS.

Element	OpenVAS	Tenable WAS
Vulnerability name	Apache HTTP Server OPTIONS Memory Leak Vulnerability (OptionsBleed)	Apache 2.4.x < 2.4.28 HTTP Vulnerability (OptionsBleed)
CVE	CVE-2017-9798	CVE-2017-9798
Description	Apache HTTP Server allows remote attackers to read data [...]	Versions of Apache 2.4.x prior to 2.4.28 are affected by a vulnerability [...]
Installed version	2.2.8	2.4.7
Fixed version	2.4.28 (or equivalent patch for 2.2.34)	2.4.28
Impact	Allows unauthorized reading of memory blocks from the server.	`https://httpd.apache.org/security/vulnerabilities_24.html\#2.4.28`
Severity (CVSS)	5.0 (Medium)	7.5 (High, CVSSv3)
Solution	Update to Apache HTTP Server 2.4.28 [...]	Update to Apache HTTP Server 2.4.28 [...]
Detection method	Apache Web Server Detection (OID: 1.3.6.1.4.1.25623.1.0.900498)	Plugin ID 98913
Family / Category	Web Server Vulnerability	Component Vulnerability
References	`http://openwall.com/lists/oss-security/2017/09/18/2`[...]	`https://httpd.apache.org/security/vulnerabilities_24.html\#2.4.28` [...]

Therefore, to construct unified and consistent datasets, a fundamental requirement for training machine learning models, we implemented an explicit field–to–field mapping in the prompt sent to the language model, aligning scanner–specific attributes with a generalized schema. This strategy enables the framework to handle the structural variability and naming heterogeneity present in reports produced by different vulnerability scanners. Missing attributes were explicitly assigned `NULL` values to avoid hallucinated content and to preserve fidelity to the original data. The complete mapping, documented in the Vulnerability Extractor specification, defines the canonical schema used to normalize fields across tools and ensures reproducible transformations.

This standardized mapping also strengthens downstream analysis by enabling cross–scanner comparisons, dataset merging, and the creation of training corpora with stable semantics, which are essential for supervised learning and vulnerability prioritization tasks. Furthermore, the LLM–based extraction approach retains applicability in operational environments where organizations receive only PDF reports from external audits or consolidated multi–scanner assessments, situations in which structured formats are often unavailable. In such scenarios, the unified schema remains valid, allowing consistent ingestion, normalization, and analysis regardless of the original report source or

level of structure.

3. Related Work

As summarized in Table 4, automated extraction of unstructured data from technical documents has become a relevant advancement in cybersecurity, especially for vulnerability identification and analysis. Recent studies demonstrate that LLMs such as GPT-4, LLaMA, and Claude improve both precision and semantic consistency in various information extraction scenarios. Works such as [Fabacher et al. 2025] show significant gains in clinical contexts, while [Li et al. 2024] and [Zhong et al. 2024] explore the potential of LLMs in visual and multimodal documents, highlighting their versatility in integrating text and images. Approaches such as [Hu et al. 2025], complemented by analyses from [Chen 2025], reinforce the models' ability to generalize extraction tasks, their high adaptability, and efficiency in cross-domain transfer.

Table 4. Information extraction with LLMs

Reference	**Domain / Application**	**Document Type**	**Model (LLMs)**
[Hu et al. 2025]	General / multiple KIE domains	Tables + non-standard layouts	GPT-4, T5-XXL, LLaMA-3
[Fabacher et al. 2025]	Clinical / multilingual medical notes	Clinical text (French and English)	GPT-4, LLaMA-3, Mistral-7B
[Zhong et al. 2024]	Multimodal / image-text	Multimodal AI documents	CLIP, BLIP-2, GPT-4V
[Chen 2025]	General AI / prompt engineering	Review / survey	GPT-4, Gemini 2, Claude 3, LLaMA-3, Mistral-Large
[Yan et al. 2025]	Technical / document processing	Technical documents and forms	T5, BART, LLaMA-2, GPT-4
This work	**Vulnerabilities / *Chunking***	**Vulnerability reports (OpenVAS / Nessus)**	**GPT-4, GPT-4.1, LLaMA-3, LLaMA-4, DeepSeek**

Despite these advances, the current literature focuses mainly on clinical, multimodal, or generic technical documents, with limited emphasis on operational security scanners. This work differs by specifically investigating standardized extraction of vulnerabilities from heterogeneous OpenVAS and Tenable WAS reports, a domain in which structural and semantic differences directly affect analysis and prioritization. Furthermore, we present explicit field mapping, integration with chunking strategies, and guidelines for generating consistent and anonymizable datasets, aspects still underexplored in the existing literature.

4. Extraction Pipeline Using LLMs

The developed tool automates the extraction of vulnerabilities from PDF reports generated by OpenVAS and Tenable WAS vulnerability scanners, using LLMs to convert unstructured text into structured representations. The main pipeline is organized into modular stages that ensure integrity and consistency of the extracted data.

As illustrated in Figure 1, the process begins with reading the report and extracting textual content while preserving fidelity to the original data. Next, the text is divided into logical blocks (chunks) to maintain the context of each vulnerability within the token limitations imposed by language models.

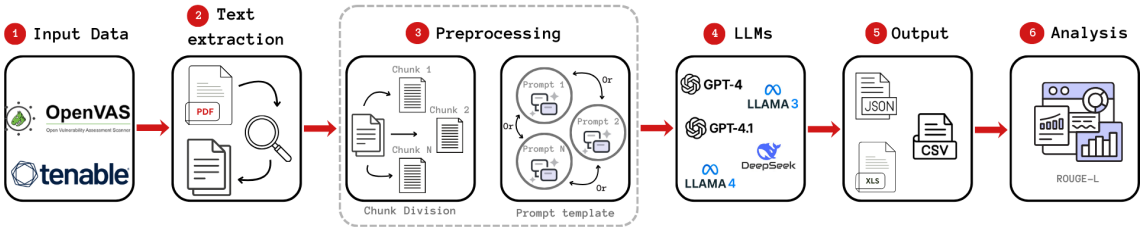

Figure 1. Extraction pipeline

Each block is processed by a specific prompt that instructs the LLM to identify relevant fields such as description, impact, solution, and references, returning a structured and semantically coherent output. This step accommodates variations across vulnerability scanners and ensures consistency in the extracted results.

Finally, the data undergo validation and consolidation that removes duplicates, checks syntactic conformity, and reconstructs the full vulnerability set. This step ensures information integrity and prepares the final output for integration with anonymization and prioritization modules.

5. Evaluation

For evaluation, we used an OpenVAS technical report containing 34 vulnerabilities of varying severities, from which a manually constructed baseline was validated by two independent analysts. Using this reference set, automated extraction experiments were conducted using GPT-4, GPT-4.1, Llama-3, Llama-4, and DeepSeek, all configured with temperature $T = 0.2$ and average blocks of approximately 9,000 characters, respecting each LLM's token limits. Similarity between extractions and the baseline was assessed using the *ROUGE-L* metric, classifying results into Divergent (≤ 0.4), Slightly Similar (≤ 0.6), Moderately Similar (≤ 0.7), and Highly Similar (> 0.7).

The results, shown in Figure 2, indicate that DeepSeek and GPT-4.1 achieved the best performance in structured extraction, demonstrating greater capacity for interpretation and preservation of original content. This result can be attributed to more recent architectures, broader training datasets, and optimizations aimed at deep contextual understanding.

In contrast to more robust models, Llama-3 and Llama-4 prioritize computational efficiency and lower operational costs, which may limit performance in complex extraction tasks requiring high semantic consistency. GPT-4 also performed worse than GPT-4.1, reflecting architectural and alignment improvements introduced in the most recent version. Qualitative inspection confirmed occasional inconsistencies such as duplications, omissions, and labeling errors, especially in similar vulnerabilities related to SSL/TLS protocols, as well as substitutions with semantically close terms that harmed exact correspondence with the baseline.

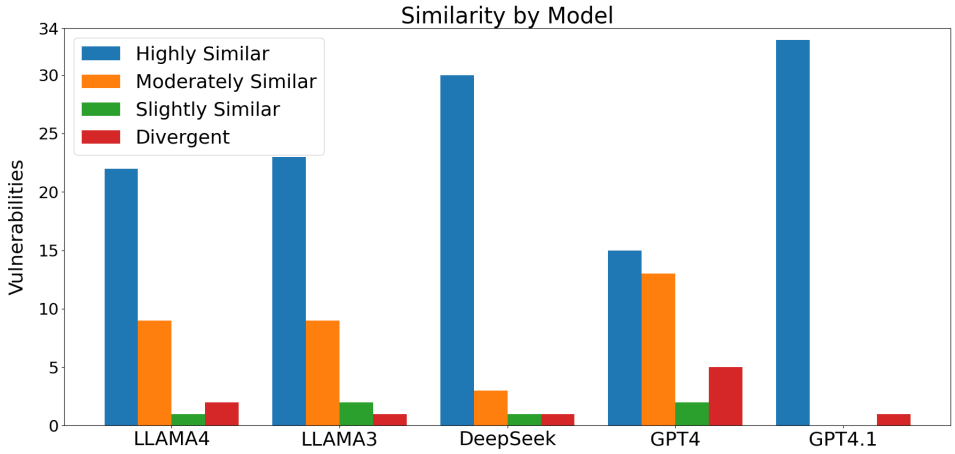

Figure 2. Average similarity (ROUGE-L) between extractions and the baseline

The analysis of precision degradation, represented by the percentage of fields with similarity below 70% in Figure 3, identified four main contributing factors: context limitations caused by chunking, which reduce global visibility of the vulnerability; semantic truncation and hallucinations triggered by cuts in technical sections such as `"NVT:"` or `"CVSS:"`; and other irregularities such as loss of delimiters and minor tokenization variations between executions. These elements explain the observed discrepancies and highlight the need for more robust segmentation and validation strategies to increase extraction reliability.

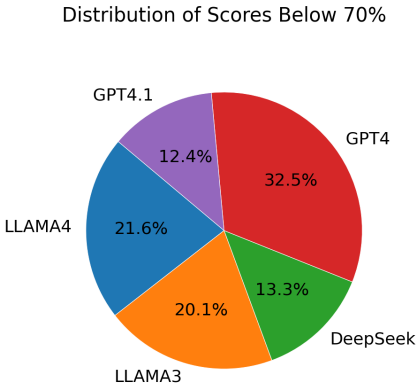

Figure 3. Similarity below Highly Similar

Delimiter loss was also identified, recurring during PDF extraction and causing shifts or fragmentations of fields such as *Summary* and *Impact*. Together with random variations arising from tokenization differences or the sampling process, even at lower temperatures, these issues produce inconsistencies between runs. These combined factors explain the observed reduction in precision and reinforce the need for adequate chunking strategies, along with post-processing validation mechanisms, to ensure the integrity and reliability of the extractions.

6. Final Considerations

This work presents an LLM-based method to extract vulnerabilities from OpenVAS and Tenable WAS reports, producing standardized datasets through explicit field mapping

and logical chunking. The approach addresses cross-scanner interoperability by normalizing heterogeneous schemas into a unified format. Evaluation shows that GPT-4.1 and DeepSeek achieve high similarity (ROUGE-L > 0.7) relative to the baseline, demonstrating effective structuring of complex technical reports.

However, the analysis identified challenges including context limitations from chunking, semantic truncation, and delimiter loss. As future work, we intend to first address these limitations by: (1) investigate robust chunking strategies with improved context preservation. Subsequently, we plan to: (2) extend evaluation to additional scanners, (3) investigate smaller language models to reduce costs, and (4) incorporate anonymization mechanisms for secure dataset sharing.

Acknowledgments

This research was partially supported by CT-Cibersegurança/RNP[2]; by CNPq[3], grant 409743/2025-9; and by FAPERGS[4], through grant agreements 24/2551-0001368-7 and 24/2551-0000726-1.

References

Chen, B. (2025). Unleashing the potential of prompt engineering for large language models. *Artificial Intelligence Review*.

Fabacher, M., Meyer, S., and Lang, A. (2025). Efficient extraction of medication information from clinical notes: An evaluation in two languages. *arXiv preprint arXiv:2502.03257*.

Greenbone (2025). Openvas report - user manual. `https://docs.greenbone.net/OPENVAS-REPORT-Manual/en/analyzing-data.html`.

Hu, Y., Li, J., and Wang, H. (2025). Large language model driven transferable key information extraction. *Scientific Reports*. Online; Acesso em 10 nov. 2025.

Li, X., Zhou, M., and Tang, Y. (2024). Enhancing visual information extraction with large language models. In *Intelligent Data Engineering and Automated Learning*. Springer.

Rede Nacional de Ensino e Pesquisa (2018). Relatório anual de segurança de 2017. Technical report, Rede Nacional de Ensino e Pesquisa, Brasil.

Rede Nacional de Ensino e Pesquisa (2024). Relatório anual de segurança de 2023. Technical report, Rede Nacional de Ensino e Pesquisa, Brasil.

Tenable (2025). Tenable web app scanning user guide. `https://docs.tenable.com/web-app-scanning/Content/PDF/Tenable_Web_App_Scanning-User_Guide.pdf`.

Yan, T., Zhang, P., and Xu, L. (2025). Docextractnet: A novel framework for enhanced document information extraction. *Information Processing & Management*.

Zhong, Z., Li, Y., and Zhang, J. (2024). Enhancing multimodal large language models with multi-instance visual prompt generator for visual representation enrichment. Amazon Science.

[2] `https://plataforma.rnp.br/ct-ciberseguranca`
[3] `https://www.gov.br/cnpq/pt-br`
[4] `https://fapergs.rs.gov.br`