Poster: LLM-Driven Exploit Assessment for Penetration Testing

Xiangmin Shen*, Wenyuan Cheng†, Yan Chen*, Zhenyuan Li†, Wencheng Zhao‡ and Dawei Sun‡
*Northwestern University, †Zhejiang University, ‡Ant Group

***Abstract*—Penetration testers face challenges in selecting effective exploits due to inconsistent quality, lack of structured evaluation, and inefficient prioritization. Existing scoring methods like CVSS and EPSS fail to assess exploit usability, while penetration testing tools offer limited guidance on exploit effectiveness. We propose an LLM-driven automated exploit assessment system that ranks exploits based on usability, reliability, and contextual applicability. Unlike previous work, which focused on vulnerability management, our system integrates into penetration testing workflows, assisting both manual and automated testing. Evaluations on 500+ exploits across 96 vulnerabilities show improved vulnerability prioritization compared to CVSS and EPSS rankings. The system enhances exploit selection efficiency, reduces manual testing overhead, and improves pentesting automation.**

I. Introduction

Penetration testing is essential for identifying exploitable vulnerabilities in systems. In real-world scenarios, pentesters—whether using manual methods or automated frameworks—must select and execute exploits from large vulnerability datasets. However, exploit selection remains a major challenge due to the variability in exploit quality, lack of structured exploit ranking, and the time-intensive nature of manual selection. Not all exploits are functional or practical, some require modifications, dependencies, or specific configurations that pentesters must manually resolve. Existing scoring systems such as CVSS and EPSS prioritize vulnerabilities but fail to assess exploit usability, success rates, or feasibility. Pentesters often rely on trial and error when selecting exploits, leading to wasted time and effort.

Existing penetration testing tools, such as Metasploit, provide exploit repositories but lack systematic exploit usability assessment. Other works provided structured exploit scoring but was designed for vulnerability management rather than real-time pentesting workflows.

To address these challenges, we propose an LLM-driven automated exploit assessment system that systematically ranks exploits based on usability, reliability, and exploitability in real-world settings. Our system integrates with pentesting workflows to improve exploit selection efficiency. We leverage LLM-based analysis to dynamically assess exploit prerequisites, execution complexity, and real-world applicability. Our system enhances both manual and automated penetration testing by reducing manual effort, improving exploit selection, and increasing testing efficiency.

II. Background and Related Work

A. The Challenge of Exploit Selection in Penetration Testing

As shown in Fig. 1, penetration testers typically follow a three-stage workflow: reconnaissance to identify potential vulnerabilities, exploit selection to choose the best available exploits, and execution and reporting to validate exploitation success. While reconnaissance is well-supported by tools such as Nmap and Nessus, and execution can be automated using tools like Metasploit, exploit selection remains largely manual, requiring pentesters to test multiple exploits before finding one that works.

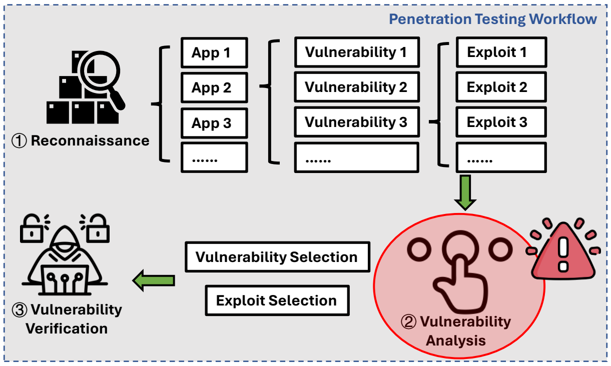


Fig. 1: An overview of Penetration Testing Workflow

B. Existing Exploit Ranking and Scoring Methods

Existing vulnerability scoring systems provide limited exploit usability assessment. CVSS ranks vulnerabilities but does not evaluate exploit reliability. EPSS predicts exploit likelihood but lacks real-world usability considerations. Metasploit's exploit ranking is manually assigned and lacks explainability. Several academic works [1], [2] aim to improve vulnerability assessment by leveraging machine learning techniques, such as NLP techniques and neural networks. However, these approaches still suffer from significant challenges related to transparency and explainability, limiting their practical utility in real-world scenarios.

C. Improvements from Our Previous Work

This poster is related to two of our papers under submission. One paper focuses on automating the penetration testing process without effective vulnerability and exploit selection. The other paper proposed a vulnerability assessment system designed for vulnerability management teams to assess vulnerability severity using LLM-based techniques. However, it did not integrate into real-world penetration testing workflows. This poster extends our previous works by bridging the gap between exploit assessment and execution in penetration testing, supporting real-time exploit selection in both manual and automated settings, and providing LLM-driven decision-making to improve efficiency.

III. System Design

As shown in Fig. 2, our LLM-driven exploit assessment system consists of three components: exploit data collection, feature extraction and analysis, and exploit usability scoring. We collect exploit data from online sources like GitHub and Google. Using LLM-based techniques, we extract key exploit characteristics, including execution reliability, target applicability, and complexity of exploitation. The system generates structured usability scores to rank exploits effectively and provides recommendations for pentesters, enabling both manual and automated selection of the best exploits. The vulnerability and exploit rankings will be used to generate vulnerability and exploit selection shown in Fig. 1.

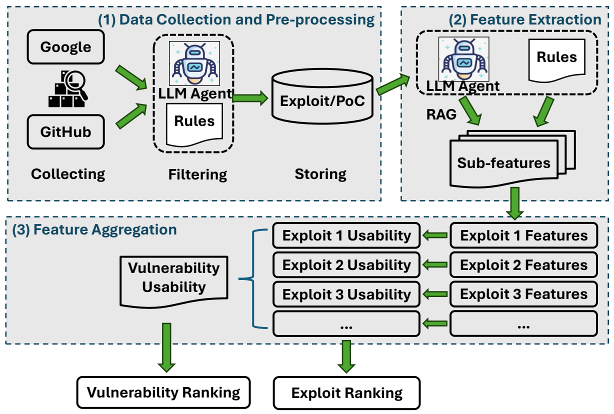
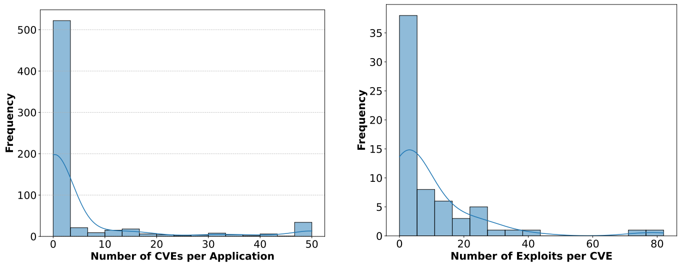


Fig. 2: An overview of Vulnerability & Exploit Assessment

IV. Preliminary Evaluation

A. Measurement Study

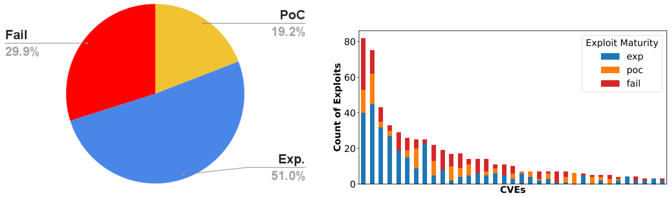
We conducted a measurement study analyzing vulnerability counts, exploit counts, and exploit maturity levels. Fig. 3 illustrates the distribution of vulnerabilities and associated exploits, showing that in over 15% of cases, an application has more than 10 potential vulnerabilities. In addition, in 35% of cases, a vulnerability have more than 10 available exploits. Fig. 4 further highlights the significant variation in exploit maturity across vulnerabilities, indicating that merely selecting a high-priority vulnerability is insufficient—choosing the right exploit is equally critical for effective penetration testing.



(a) Distribution of CVE Counts per Application

(b) Distribution of Exploits Counts per CVE

Fig. 3: Measurement on Vulnerability and Exploit Counts



(a) Overall Distribution of Exploit Maturity

(b) Distribution of Exploit Maturity per CVE

Fig. 4: Measurement on Exploit Maturity

B. Exploit Usability

We evaluate our system using 55 vulnerabilities which have over 300 exploits available online. Three key metrics were used in this evaluation:

- Top-k Success Rate, which measures whether at least one exploit achieving the intended functionality appears in the top-k recommendations;
- Precision@k, which evaluates whether the highest-quality exploit is included in the top-k recommendations;
- Recall@k for Top-j, which assesses whether at least one of the top-j exploits is included in the top-k recommendations.

Our results show that 83% of top-ranked exploits successfully executed, validating the effectiveness of our ranking approach. Additionally, 100% of the top three recommendations contained at least one successful exploit, ensuring strong reliability. The highest-ranked exploit appeared in the top 3 recommendations 58% of the time, while at least one of the top 3 exploits appeared in the top 3 recommendations 75% of the time. Our results demonstrate the system's effectiveness in prioritizing high-quality exploits

C. Vulnerability Usability

We collaborate with a leading red team of Ant Group to evaluate our vulnerability selection. They provided 96 vulnerabilities with 500+ exploits, and we evaluated their usability. The pentesters confirmed our usability evaluation results on 91 out of 96 vulnerabilities, outperforming EPSS (65 out of 96 confirmed) significantly.

V. Conclusion and Future Work

We propose an LLM-driven automated exploit assessment system that enhances penetration testing workflows by improving vulnerability and exploit selection. Our system bridges vulnerability assessment with real-world pentesting, benefiting both human testers and automated tools. Future work includes expanding the dataset to improve exploit ranking accuracy, refining scoring models using real-world pentesting feedback, and integrating reinforcement learning for adaptive exploit selection.

References

[1] T. H. Le, H. Chen, and M. A. Babar, "A survey on data-driven software vulnerability assessment and prioritization," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–39, 2022.

[2] J. Yin, G. Chen, W. Hong, H. Wang, J. Cao, and Y. Miao, "Empowering vulnerability prioritization: A heterogeneous graph-driven framework for exploitability prediction," in *International Conference on Web Information Systems Engineering*. Springer, 2023, pp. 289–299.

Poster: LLM-Driven Exploit Assessment for Penetration Testing

Xiangmin Shen[1], Wenyuan Cheng[2], Yan Chen[1], Zhenyuan Li[2], Wencheng Zhao[3], Dawei Sun[3]
[1]Northwestern University, [2]Zhejiang University, [3]Ant Group

Motivation

Penetration testing relies heavily on selecting effective exploits, but this process is often inefficient due to:

- **Inconsistent Exploit Quality:** Not all exploits perform equally, leading to variable outcomes.
- **Lack of Structured Ranking:** Existing tools do not provide standardized usability rankings.
- **Manual Selection Challenges:** Current methods require time-consuming manual assessments.

Challenges with Existing Methods

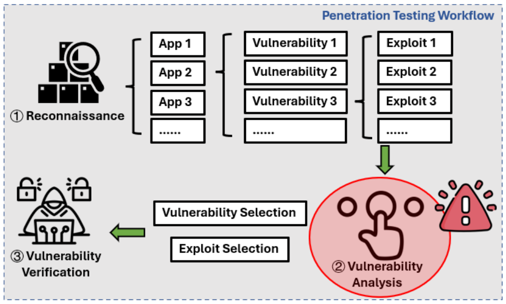
- **Lack of Usability Assessment:** While they rank vulnerabilities, they do not assess exploit usability.
- **Proprietary:** Platforms like Metasploit and Tenable offer limited exploits with non-explainable rankings.

Our Contribution

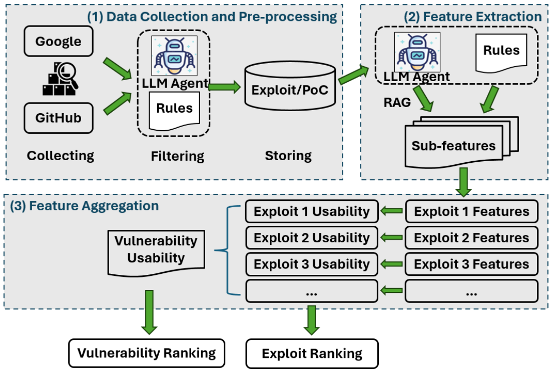
- **Systematic Exploit Ranking:** an LLM-driven framework to rank exploits based on real-world usability.
- **Seamless Integration:** Compatible with both manual and automated penetration testing workflows.
- **Improved Efficiency:** Reduces the time and effort needed for effective exploit selection.

Framework

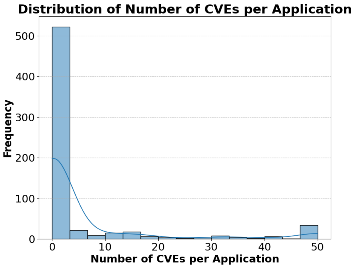
Penetration Testing

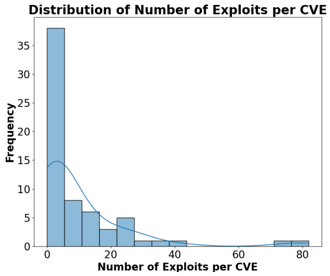


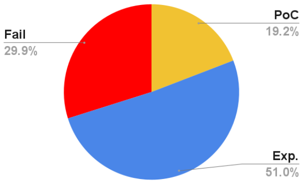
Vulnerability & Exploit Analysis

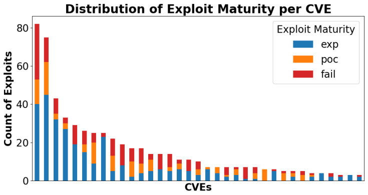


Measurement Study









Key Results

- 15% (97 out of 655) of the applications have more than 10 associated CVEs
- 35% (19 out of 55) of the vulnerabilities have more than 10 available exploits
- Exploit maturity varies among CVEs

Preliminary Evaluation

Exploit Usability

- Dataset: 55 vulnerabilities, 300+ exploits.
- Metrics
  - **Top-k Success Rate**: Measures if at least one effective exploit appears in the top-k recommendations;
  - **Precision@k**: Checks if the highest-quality exploit is among the top-k recommendations;
  - **Recall@k for Top-j:** Assesses if top-j exploits are included within top-k recommendations.
- Results on 8 vulnerabilities, 60+ exploits

Metric	Value
Top-1 Success Rate	83.3%
Top-3 Success Rate	100%
Precision @ 3	58.3%
Recall @ 3 for Top-3	75.0%

Vulnerability Usability

- Dataset: 96 vulnerabilities, 500+ exploits.
- Method: Manual verification of vulnerability usability scoring by experienced pentesters
- Results:
  - Our accuracy: 94.8% (91 out of 96)
  - EPSS accuracy: 67.7% (65 out of 96)

Cost Analysis

- Dataset: 96 vulnerabilities, 500+ exploits.
- Results:

Metric	w/ GPT-4o	w/GPT-3.5
Avg. Analysis Time	54.67s	40.54s
Avg. Cost	0.11 USD	0.03 USD
Avg. Token Usage	347795.8	359284.4

Takeaways

- **Effective Prioritization is Crucial:** The large number of available vulnerabilities and exploits necessitates structured ranking, as their exploit maturity varies significantly and impacts selection efficiency.
- **High Accuracy with Practical Efficiency:** Our approach improves usability assessment accuracy while maintaining reasonable time and cost efficiency.
- **Enhanced Decision-Making:** Our framework enables practitioners to quickly identify and select effective exploits.

The QR code for Xiangmin's homepage: https://nbshenxm.github.io/