USENIX Security '24 Artifact Appendix: PENTESTGPT: Evaluating and Harnessing Large Language Models for Automated Penetration Testing

Gelei Deng
Nanyang Technological University

A Artifact Appendix

We introduce PENTESTGPT, an LLM-empowered automated penetration testing framework that leverages the abundant domain knowledge inherent in LLMs. PENTESTGPT is meticulously designed with three self-interacting modules, each addressing individual sub-tasks of penetration testing, to mitigate the challenges related to context loss. Our evaluation shows that PENTESTGPT not only outperforms LLMs with a task-completion increase of 228.6% compared to the GPT-3.5 model among the benchmark targets, but also proves effective in tackling real-world penetration testing targets and CTF challenges. Having been open-sourced on GitHub, PENTESTGPT has garnered over 6,500 stars in 12 months and fostered active community engagement, attesting to its value and impact in both the academic and industrial spheres.

A.1 Abstract

PENTESTGPT is an LLM-empowered automatic penetration testing tool. It works on an interactive mode: the tester queries the PENTESTGPT directly in natural language regarding the next-step in penetration testing, similar to using ChatGPT. PENTESTGPT performs the reasoning process through LLMs, and generate a concrete step to follow so that the tester can execute in the testing environment. In this way, users with no prior expert knowledge can perform penetration testing.

A.2 Description & Requirements

Given the complexities involved in the design of PENTESTGPT and its application in penetration testing, fully reproducing results from the original study presents several challenges: (1) the outputs of LLMs are inherently non-deterministic, (2) proficient knowledge in penetration testing is essential to effectively utilize PENTESTGPT, and (3) certain benchmark machines are exclusive to HackTheBox subscribers. This section will first outline the proper usage of the tool, followed by instructions for a sample run. We will also provide video demonstrations and tool logs to validate the reproducibility of results. As PENTESTGPT is an open-source project, we encourage further exploration through its GitHub repository and associated discussion groups.

A.2.1 Security, privacy, and ethical concerns

It is important to consider several key points regarding security and privacy:

1. PENTESTGPT interacts with the OpenAI API, hence interactive data is retained by OpenAI for 30 days. Utilizing PENTESTGPT on real-world systems outside of designed benchmarks could lead to data leakage and privacy issues.
2. Inherent risks in penetration testing could result in system or network damage if operations are mistakenly conducted on incorrect targets.
3. As an automated tool, PENTESTGPT has the potential for misuse in unauthorized or malicious activities. To mitigate this, we have integrated an ethical use declaration in the project's open-source repository.

A.2.2 How to access

PENTESTGPT is primarily hosted on GitHub at https://github.com/GreyDGL/PentestGPT. For this artifact review, a specific version has been archived on Zenodo at https://zenodo.org/record/12260307.

A.2.3 Hardware Requirements

PENTESTGPT is compatible with any modern PC or laptop capable of running virtual machines via VirtualBox.

A.2.4 Software Requirements

PENTESTGPT is developed in Python3 and is accessible online as an open-source tool. Setup is straightforward with virtual environments via *pip*. We recommend Python version 3.10, which has undergone extensive testing. While PENTESTGPT is optimized for the latest version of MacOS and Ubuntu 22.04, it is also compatible with Unix-based systems oriented towards penetration testing, such as Parrot and Kali Linux. Running PENTESTGPT effectively requires a valid OpenAI API key. For practical penetration testing, it is advisable to simulate a target machine as described in the original documentation, typically using a VirtualBox VM with at least 1GB RAM and a single-core CPU.

A.2.5 Benchmarks

Due to the nature of penetration testing and PENTESTGPT, it is difficult to reproduce all the experimental results because (1) users need to manually execute the complete penetration testing process, and (2) users need to accurately describe the findings and thoughts towards the execution results as instructed by PENTESTGPT. In the artifact README file, we include a complete benchmark we used in the experiments, which are all open-source available online. We also include a detailed set up instruction for one particular benchmark machine so that you could experiment PENTESTGPT on it.

A.3 Set-up

A.3.1 Installation

Follow *READEME.md*, a Python 3.10 virtual environment shall be created. Users can then install the tool directly by pulling from Github or local installation. Users shall then export the OpenAI API key to the system environment, and the installation is completed.

A.3.2 Basic Test

Once the installation is completed, users can run `pentestgpt-connection` to test if the tool is properly installed, and connected to the OpenAI API.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): PENTESTGPT could instruct the users to perform penetration testing, without any external expert knowledge. Users can use them as a chatbot, and complete end-to-end penetration testing by following the instructions, where as the original GPT models cannot complete it.

A.4.2 Experiments

We provide the complete instruction for users to complete one sample experiment over the target benchmark machine. More details are included in the README file. Considering that some artifact reviewers may not have penetration testing experiences, we have been uploading YouTube videos to demonstrate the whole process in the past, with some more examples on how PENTESTGPT works on real-world targets. For the general usage of PENTESTGPT, please refer to to https://www.youtube.com/@geleideng5744. We will upload a new video to demonstrate the following process in details.

(E1): *[Conduct the penetration testing on Target Hackable II with* PENTESTGPT*] [1hr human-minutes]:*

How to: Install the Hackable II pentesting target through virtual machine. Install PENTESTGPT and test connection. Run PENTESTGPT over the target by asking questions and following the instructions, until the machine is completed. This machine is reported to have 6 success among 10 tests when 2 certified penetration testers are operating on it with PentestGPT.

Preparation: Install the Hackable II pentesting target through virtual machine (VirtualBox is recommended. This target is open-sourced and more details are available at https://www.vulnhub.com/entry/hackable-ii,711/. Install PENTESTGPT follow the instruction in the tool repository.

Execution: To conduct penetration testing with the help of PENTESTGPT, simply open a separate terminal and follow the given instructions. This starts from providing the IP address of the installed penetration testing target to PENTESTGPT. It will provide concrete instructions to follow (typically starting from an `nmap` scan. Execute the instruction in the terminal, and paste the commands back to PENTESTGPT, so that it will reason the next step. Repeat this process. Note that PENTESTGPT may some time require you to look up for online exploits or hack information. You may copy-paste the complete website source code, or google search results back to PENTESTGPT for it conduct further reasoning.

Results: Ideally, you should be able to complete penetration testing on the target machine Hackable. An official walkthrough of the correct approach is documented at https://medium.com/@networkdavit/hackable-ii-vulnhub-walkthrough-9be05d19e4f1. However, PENTESTGPT can introduce false steps to explore, which is very common in the penetration testing process as human testers will also explore on false directions that may not lead to the final exploitation.

(E1): *[Conduct Penetration Testing on Hackable II with ChatGPT]:*

How to: Repeat the same process with ChatGPT GPT-4. Unfortunately we cannot reproduce what we used before as ChatGPT GPT-4 now only supports newer model, not the 32k token size one we used in the paper. However, you may still find that the latest model cannot complete the pentest process.

A.5 Notes on Reusability

The project has been developed in open-source status in the past one year, and any comments/issues/discussions are welcomed. Note that with the changes on OpenAI APIs, the LLM tested and illustrated in the original manuscript may not be available in the future.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2024/.