PentestGPT

A GPT-empowered penetration testing tool.
Explore the docs »

Design Details · View Demo · Report Bug or Request Feature

General Updates

- [Update on 25/10/2024] We're completing the refactoring of PentestGPT and will release v1.0 soon!
- [Update on 12/08/2024] The research paper on PentestGPT is published at USENIX Security 2024
- [Update on 25/03/2024] We're working on the next version of PentestGPT, with online searching, RAGs and more powerful prompting. Stay tuned!
- [Update on 17/11/2023] GPTs for PentestGPT is out! Check this: https://chat.openai.com/g/g-4MHbTepWO-pentestgpt
- [Update on 07/11/2023] GPT-4-turbo is out! Update the default API usage to GPT-4-turbo.
- Available videos:
  - The latest installation video is here.
  - **PentestGPT for OSCP-like machine: HTB-Jarvis**. This is the first part only, and I'll complete the rest when I have time.
  - **PentestGPT on HTB-Lame**. This is an easy machine, but it shows you how PentestGPT skipped the rabbit hole and worked on other potential vulnerabilities.
- **We're testing PentestGPT on HackTheBox**. You may follow this link. More details will be released soon.
- Feel free to join the Discord Channel for more updates and share your ideas!

Quick Start

1. Create a virtual environment if necessary. (`virtualenv -p python3 venv` , `source venv/bin/activate`)
2. Install the project with `pip3 install git+https://github.com/GreyDGL/PentestGPT`
3. **Ensure that you have link a payment method to your OpenAI account.** Export your API key with `export OPENAI_API_KEY='<your key here>'` ,export API base with `export OPENAI_BASEURL='https://api.xxxx.xxx/v1'` if you need.
4. Test the connection with `pentestgpt-connection`
5. For Kali Users: use `tmux` as terminal environment. You can do so by simply run `tmux` in the native terminal.
6. To start: `pentestgpt --logging`

Getting Started

- **PentestGPT** is a penetration testing tool empowered by **ChatGPT**.
- It is designed to automate the penetration testing process. It is built on top of ChatGPT and operate in an interactive mode to guide penetration testers in both overall progress and specific operations.
- **PentestGPT** is able to solve easy to medium HackTheBox machines, and other CTF challenges. You can check this example in `resources` where we use it to solve HackTheBox challenge **TEMPLATED** (web challenge).
- A sample testing process of **PentestGPT** on a target VulnHub machine (Hackable II) is available at here.
- A sample usage video is below: (or available here: Demo)

Common Questions

- **Q**: What is PentestGPT?

    - **A**: PentestGPT is a penetration testing tool empowered by Large Language Models (LLMs). It is designed to automate the penetration testing process. It is built on top of ChatGPT API and operate in an interactive mode to guide penetration testers in both overall progress and specific operations.
- **Q**: Do I need to pay to use PentestGPT?
    - **A**: Yes in order to achieve the best performance. In general, you can use any LLMs you want, but you're recommended to use GPT-4 API, for which you have to link a payment method to OpenAI.
- **Q**: Why GPT-4?
    - **A**: After empirical evaluation, we find that GPT-4 performs better than GPT-3.5 and other LLMs in terms of penetration testing reasoning. In fact, GPT-3.5 leads to failed test in simple tasks.
- **Q**: Why not just use GPT-4 directly?
    - **A**: We found that GPT-4 suffers from losses of context as test goes deeper. It is essential to maintain a "test status awareness" in this process. You may check the PentestGPT Arxiv Paper for details.
- **Q**: Can I use local GPT models?
    - **A**: Yes. We support local LLMs with custom parser. Look at examples here.

Installation

PentestGPT is tested under `Python 3.10`. Other Python3 versions should work but are not tested.

Install with pip

PentestGPT relies on **OpenAI API** to achieve high-quality reasoning. You may refer to the installation video here.

1. Install the latest version with `pip3 install git+https://github.com/GreyDGL/PentestGPT`
    - You may also clone the project to local environment and install for better customization and development
        - `git clone https://github.com/GreyDGL/PentestGPT`
        - `cd PentestGPT`
        - `pip3 install -e .`
2. To use OpenAI API
    - **Ensure that you have link a payment method to your OpenAI account.**
    - export your API key with `export OPENAI_API_KEY='<your key here>'`
    - export API base with `export OPENAI_BASEURL='https://api.xxxx.xxx/v1'` if you need.
    - Test the connection with `pentestgpt-connection`
3. To verify that the connection is configured properly, you may run `pentestgpt-connection`. After a while, you should see some sample conversation with ChatGPT.
    - A sample output is below

```
 You're testing the connection for PentestGPT v 0.11.0
#### Test connection for OpenAI api (GPT-4)
1. You're connected with OpenAI API. You have GPT-4 access. To start PentestGPT, please use <pentestgpt --reasoning_model=gpt-

#### Test connection for OpenAI api (GPT-3.5)
2. You're connected with OpenAI API. You have GPT-3.5 access. To start PentestGPT, please use <pentestgpt --reasoning_model=gp
```

    - notice: if you have not linked a payment method to your OpenAI account, you will see error messages.
4. The ChatGPT cookie solution is deprecated and not recommended. You may still use it by running `pentestgpt --reasoning_model=gpt-4 --useAPI=False`.

Build from Source

1. Clone the repository to your local environment.
2. Ensure that `poetry` is installed. If not, please refer to the poetry installation guide.
3.

Usage

1. **You are recommended to run**:
    - (recommended) - `pentestgpt --reasoning_model=gpt-4-turbo` to use the latest GPT-4-turbo API.
    - `pentestgpt --reasoning_model=gpt-4` if you have access to GPT-4 API.
    - `pentestgpt --reasoning_model=gpt-3.5-turbo-16k` if you only have access to GPT-3.5 API.
2. To start, run `pentestgpt --args`.
    - `--help` show the help message
    - `--reasoning_model` is the reasoning model you want to use.
    - `--parsing_model` is the parsing model you want to use.
    - `--useAPI` is whether you want to use OpenAI API. By default it is set to `True`.
    - `--log_dir` is the customized log output directory. The location is a relative directory.
    - `--logging` defines if you would like to share the logs with us. By default it is set to `False`.
3. The tool works similar to *msfconsole*. Follow the guidance to perform penetration testing.
4. In general, PentestGPT intakes commands similar to chatGPT. There are several basic commands.
    1. The commands are:
        - `help`: show the help message.
        - `next`: key in the test execution result and get the next step.

- `more`: let **PentestGPT** to explain more details of the current step. Also, a new sub-task solver will be created to guide the tester.
- `todo`: show the todo list.
- `discuss`: discuss with the **PentestGPT**.
- `google`: search on Google. This function is still under development.
- `quit`: exit the tool and save the output as log file (see the **reporting** section below).

2. You may always use `TAB` to autocomplete the commands.
3. When you're given a drop-down selection list, you can use cursor or arrow key to navigate the list. Press `ENTER` to select the item. Similarly, use <SHIFT + right arrow> to confirm selection.
   The user can submit info about:
   - **tool**: output of the security test tool used
   - **web**: relevant content of a web page
   - **default**: whatever you want, the tool will handle it
   - **user-comments**: user comments about PentestGPT operations

5. In the sub-task handler initiated by `more`, users can execute more commands to investigate into a specific problem:

   1. The commands are:
      - `help`: show the help message.
      - `brainstorm`: let PentestGPT brainstorm on the local task for all the possible solutions.
      - `discuss`: discuss with PentestGPT about this local task.
      - `google`: search on Google. This function is still under development.
      - `continue`: exit the subtask and continue the main testing session.