Vault

Dec 30, 2020

Contents

CHAPTER 1

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

CHAPTER 2

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

CHAPTER 3

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

CHAPTER 4

Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

CHAPTER 5

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at abhishek_official@hotmail.com. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

CHAPTER 6

Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at https://www.contributor-covenant.org/version/1/4/code-of-conduct.html

For answers to common questions about this code of conduct, see contributor-covenant.org/faq.

CHAPTER 7

Contributing

Thank you for your interest in contributing to our project. Whether it's a bug report, new feature, correction, or additional documentation, we greatly value feedback and contributions from our community.

Please read through this document before submitting any issues or pull requests to ensure we have all the necessary information to effectively respond to your bug report or contribution.

7.1 Reporting Bugs/Feature Requests

We welcome you to use the GitHub issue tracker to report bugs or suggest features.

When filing an issue, please check existing open, or recently closed issues to make sure somebody else hasn't already reported the issue. Please try to include as much information as you can. Details like these are incredibly useful:

- A reproducible test case or series of steps
- The version of our code being used
- Any modifications you've made relevant to the bug
- Anything unusual about your environment or deployment

7.2 Contributing via Pull Requests

Contributions via pull requests are much appreciated. Before sending us a pull request, please ensure that:

1. You are working against the latest source on the *master* branch.
2. You check existing open, and recently merged, pull requests to make sure someone else hasn't addressed the problem already.
3. You open an issue to discuss any significant work - we would hate for your time to be wasted.

To send us a pull request, please:

1. Fork the repository.
2. Modify the source; please focus on the specific change you are contributing. If you also reformat all the code, it will be hard for us to focus on your change.
3. Commit to your fork using clear commit messages.
4. Send us a pull request, answering any default questions in the pull request interface.

GitHub provides additional document on forking a repository and creating a pull request.

7.3 Finding contributions to work on

Looking at the existing issues is a great way to find something to contribute on. As our projects, by default, use the default GitHub issue labels ((enhancement/bug/duplicate/help wanted/invalid/question/wontfix), looking at any good first issue' issues is a great place to start.

7.4 Licensing

See the LICENSE file for our project's licensing. We will ask you confirm the licensing of your contribution.

CHAPTER 8

Installation

8.1 Clone the repository

Check out the code:

```
$ git clone https://github.com/abhisharma404/vault
Cloning into 'vault'...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 1323 (delta 37), reused 65 (delta 26), pack-reused 1237
Receiving objects: 100% (1323/1323), 726.16 KiB | 250.00 KiB/s, done.
Resolving deltas: 100% (738/738), done.
```

8.2 Setup a virtual environment

Format: `python3 -m venv <name-of-the-virtualenv>`

Next you need to create your virtual environment:

```
$ python3 -m ven venv
```

8.3 Install python dependencies

- Activate the virtual environment and install packages:

  ```
  $ source venv/bin/activate
  ```

- Install python dependencies:

```
$ pip install -r requirements.txt
```

CHAPTER 9

License

MIT License

Copyright (c) [2018] [Abhishek Sharma]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 10

Welcome to vault's documentation!

10.1 Quick-Start

Vault, a Swiss army knife for hackers.

It can be used for performing various kind of vulberability scans and attack on the provided host.

10.1.1 Features

- **Scan website for the following vulnerabilities**
  - XSS
  - LFI
- **Scanner**
  - Port scanning : ACK, FIN, NULL, XMAS
  - IP scanning : Ping Sweep, ARP
  - SSL vulnerability scan
  - OS scan
  - Hash scanner : MD5, SHA1, SHA224, SHA256, SHA512
- **Crawling**
  - Crawl a website and collect all the links
  - Crawl and scrape the website for images
- **Attacks**
  - DDoS Attack
  - ARP Spoofer

  - De-authentication attack
  - Ping of death
  - MAC Flood attack
- **Finder**
  - Find comments in source code
  - Find e-mails in source code
- **Others**
  - Information Gathering
  - Clickjacking
  - jQuery version checking
  - Insecure cookie flags
  - Testing HTTP methods
  - Insecure headers
  - Header/banner grabbing
  - Brute force login through authorization headers
  - URL Fuzzer
  - WHOIS Lookup
  - Google Dork
  - Admin panel finder
  - Open redirect vulnerability
  - CMS Detection
  - Detect Honeypots
- **Utilities**
  - Keylogger
  - MAC address changer

10.1.2 Requirements

Software required:

- Python 3 or above
- python-virtualenv

For Python dependencies, see requirements.txt.

10.1.3 Gitter

You can join the Vault community at the following Gitter chat room.

10.2 Parameters

- **-u or --url** This argument is to provide the URL that is to be tested. Ex: `python vault.py -u http://example.com`
- **-ip or --ip** This argument can used to provide ip that is to be scanned. Ex: `python vault.py -ip 127.0.0.1`
- **-source_port** Specifiy the source port that should be used for sending all the packets.
- **-t or --threads** This argument can be used to define the number of threads to be used while performing all the checks.
- **-interval** This argument is used to give an interval of specific time for sending packets.
- **-mac_flood** This argument can be used to change the mac address of the given interface.
- **-all** This argument is used to run all the scan that are available in Vault.

10.2.1 Arguments that can only be used with -u or -url

- **-ssl** This argument can be used to perform SSL scan. This let you scan the target and list all SSL protocols and will show you if the target is vulnerable to any of SSL vulnerabilities
- **-info** This argument will perform basic information gathering checks on the given target. The output from this can include HTTP methods used, Check if any insecure cookies are used or any insecure headers are present.
- **-comment** This argument can be used to check if there are any comments present on the given URL.
- **-fuzz** This argument can be used to peform fuzzing on the given URL. For fuzzing the payloads are used from the fuzz_payloads.txt file and that can be updated with custom payloads.
- **-email** This argument can be used to check if any related email can be found.
- **-xss** This argument is used to scan the target for XSS vulnerabilties. All the XSS payloads that are used during the scan are present in xss_payloads.txt.
- **-lfi** This argument is used to Scan target for any LFI vulnerabilties. All the LFI payloads that are used during the scan are present in lfi_payloads.txt.
- **-admin** This argument is used to find admin panel on the given target. This scan use predefined locations to check for admin panel. All the location used during the scan are present in admin_payloads.txt.
- **-orv** This argument is used to find the open redirect vulnerability in the given target. Payload used for the scan is present in orv_payloads.txt.
- **-jqeury** This argument is used to check the jQuery version used on the given target and list out all the vulnerabilities related to that version, if any.
- **-bruteforce** This argument is used to bruteforce logins on the given tagret. With this `-username` argument has to be provided. The passwords used for brute force are taken from 10k-most-common-passwords.txt.
- **cr** This argument is used to extract all the link from a webpage
- **cri** This argument is used to extract all the images from a webpage
- **-detect_cms** This argument is used to detect the CMS version the given target is running.

10.2.2 Arguments that can only be used with -ip

- **-p or --port** This can used to provide a single port for port scanning.
- **-sp or --start_port and -ep or --end_port** These arguments can be used to define the range of port for scanning. Ex: `python vault.py -ip 127.0.0.1 -sp 9000 -ep 10000`
- **whois** This can be used to perform a basic whois scan on the given IP. Ex: `python vault.py -ip 127.0.0.1 -whois`
- **-ping_sweep** This argument is used for performing ping sweeps to map s.
- **-honey** This argument is used to check if the given IP is a honeypot or not. Ex: `python vault.py -ip 127.0.0.1 -honey`

10.2.3 Arguments that can be used when the tool is used with SUDO privileages

- **-xmas** This argument can be used to perform xmas scan on the given IP
- **-fin** This argument can be used to perform fin scan on the given IP
- **-null** This argument can be used to perform null scan on the given IP
- **-ack** This argument can be used to perform ack scan on the given IP

10.2.4 Attacks

- **-arp** This argument can be used to perform arp spoofing

  Required argument: `-ip`
- **-ping_death** This argument is used to perform the ping of death attack.

  Required arguments: This attack works with both `-ip` or `-u` arguments.
- **-deauth** This argument is used to perform the deauthentication attack.

  Optional arguments:

  - `-i` or `--interface`
  - `-target_bssid`
- **-ddos** This argument is used to perform DDOS attack.

  Required argument: `-ip` or `-u`

  Optional arguments:

  - `-sp`
  - `-ep`
  - `-t`
  - `-interval`

CHAPTER 11

Indices and tables

- genindex
- modindex
- search