

User Manual for EtherAttack

- 1) Visit <https://github.com/Xenomii/EtherAttack> to view EtherAttack Project
- 2) Clone the EtherAttack project on GitHub through this link below:

<https://github.com/Xenomii/EtherAttack>

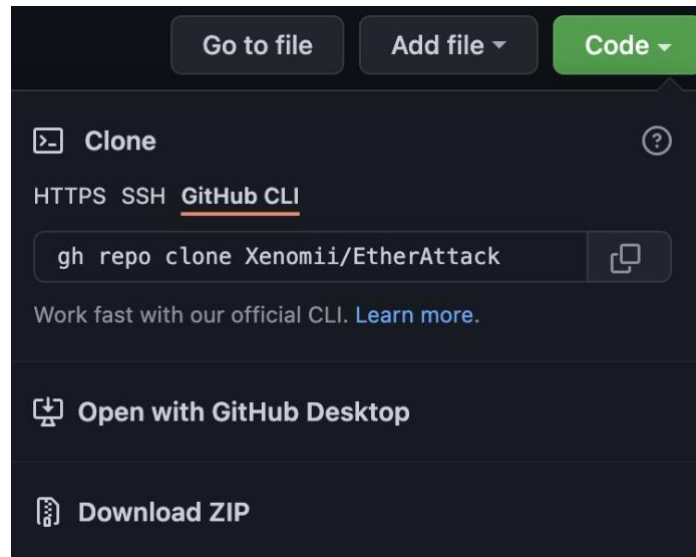


Figure 1 Clone from GitHub

- 3) Open terminal and type:

\$ git clone <https://github.com/Xenomii/EtherAttack>

```
Last login: Sun Apr  3 15:21:23 on ttys003  
brilliantstar@Stellars-MacBook-Pro ~ % git clone https://github.com/Xenomii/EtherAttack
```

Figure 2 Git clone on Terminal

- 4) Press Enter to clone the project.

```
etherattack
Cloning into 'EtherAttack'...
remote: Enumerating objects: 479, done.
remote: Counting objects: 100% (479/479), done.
remote: Compressing objects: 100% (334/334), done.
remote: Total 479 (delta 268), reused 325 (delta 136), pack-reused 0
Receiving objects: 100% (479/479), 29.89 MiB | 7.05 MiB/s, done.
Resolving deltas: 100% (268/268), done.
```

Figure 3 EtherAttack successfully cloned

- 5) Enter the path of the where EtherAttack is located

Eg: cd Path_Of_EtherAttack

```
brilliantstar@Stellars-MacBook-Pro ~ % cd /Users/brilliantstar/GitHub/EtherAttack
```

Figure 4 locate EtherAttack

- 6) Cd to webapp

Eg: cd webapp

```
brilliantstar@Stellars-MacBook-Pro EtherAttack % cd webapp
```

Figure 5 cd webapp

- 7) Install python virtual environment

sudo apt install python3-pip python3-venv

- 8) Create a python virtual environment by inputting the command:

python3 -m venv venv

. venv/bin/activate

```
brilliantstar@Stellars-MacBook-Pro webapp % python3 -m venv venv
brilliantstar@Stellars-MacBook-Pro webapp % . venv/bin/activate
```

Figure 6 create virtual environment

9) (.venv) will appear at the start of the directory terminal output.

```
[(venv) brilliantstar@Stellars-MacBook-Pro webapp %
```

Figure 7 (.venv) appearing at the start of directory terminal

10) Install dependencies that is required for this project using pip:

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

```
pip install Flask
```

```
(venv) brilliantstar@Stellars-MacBook-Pro webapp % pip install -r requirements.txt
```

```
(venv) brilliantstar@Stellars-MacBook-Pro webapp % pip install Flask  
Requirement already satisfied: Flask in /venv/lib/python2.10/site-packages
```

Figure 8 pip install

11) Install solidity compiler:

```
solc-select install 0.6.10
```

```
(venv) brilliantstar@Stellars-MacBook-Pro webapp %  
(venv) brilliantstar@Stellars-MacBook-Pro webapp % solc-select install 0.6.10
```

Figure 9 install solidity compiler

12) Verify that the compiler is installed

```
solc-select versions
```

```
version "0.6.10" installed.  
(venv) brilliantstar@Stellars-MacBook-Pro webapp % solc-select versions  
0.6.10 (current, set by /Users/brilliantstar/.solc-select/global-version)
```

Figure 10 verify that compiler is well installed

13) Run the application:

```
python3 app.py
```

```
[(venv) brilliantstar@Stellars-MacBook-Pro webapp % python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 295-929-250
```

Figure 11 run the application

14) Visit <http://127.0.0.1:5000> . EtherAttack webpage would be shown

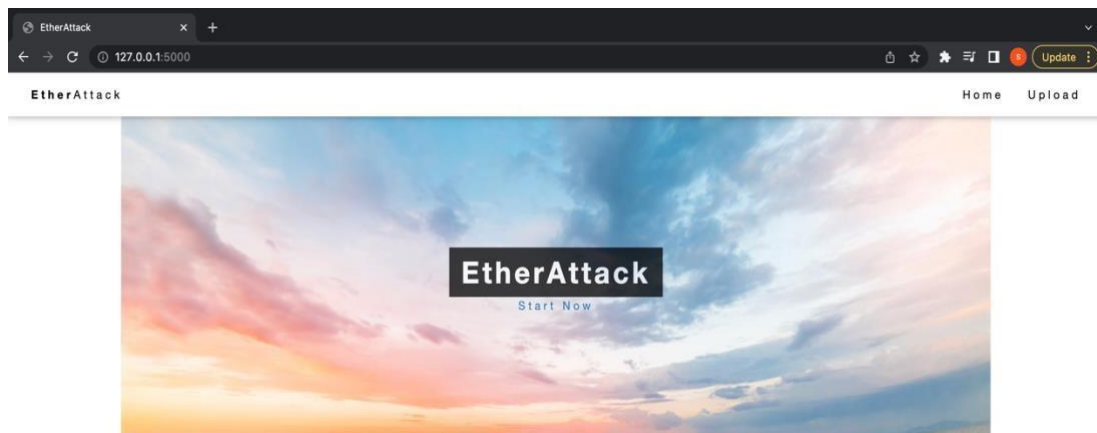
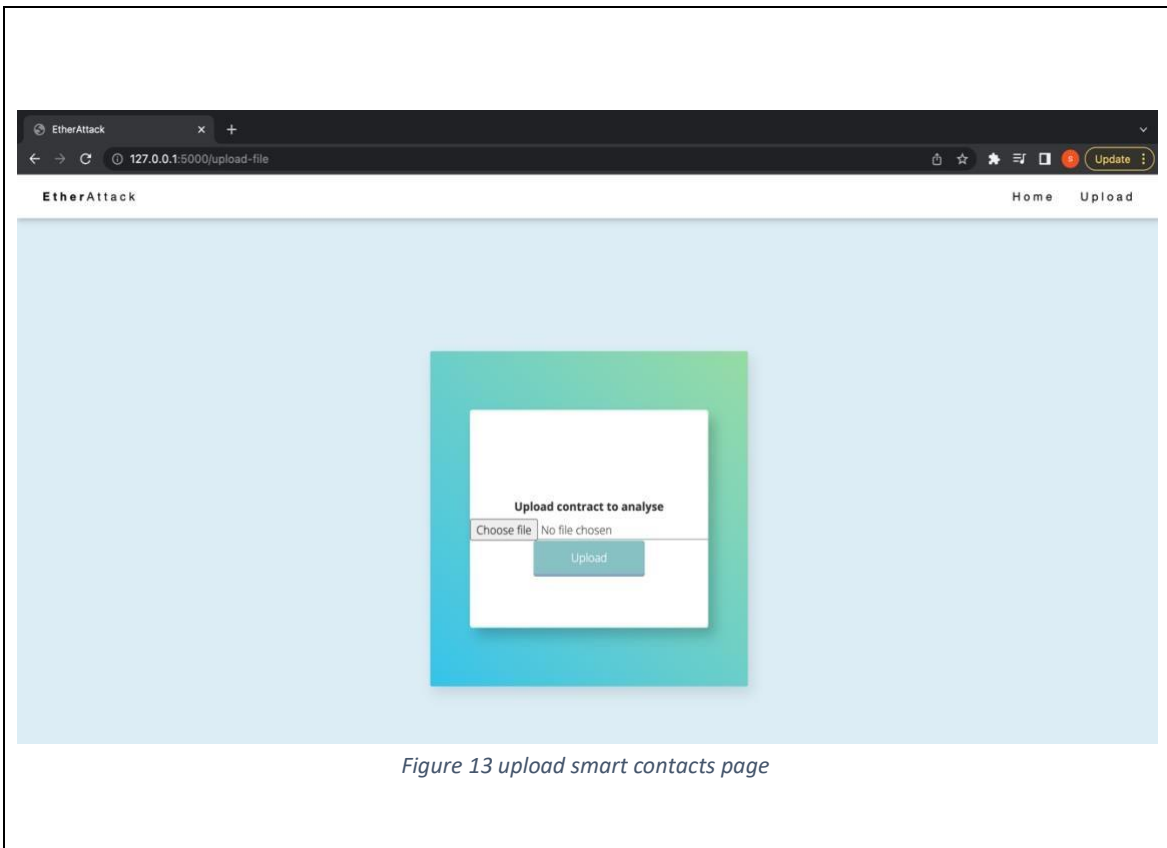
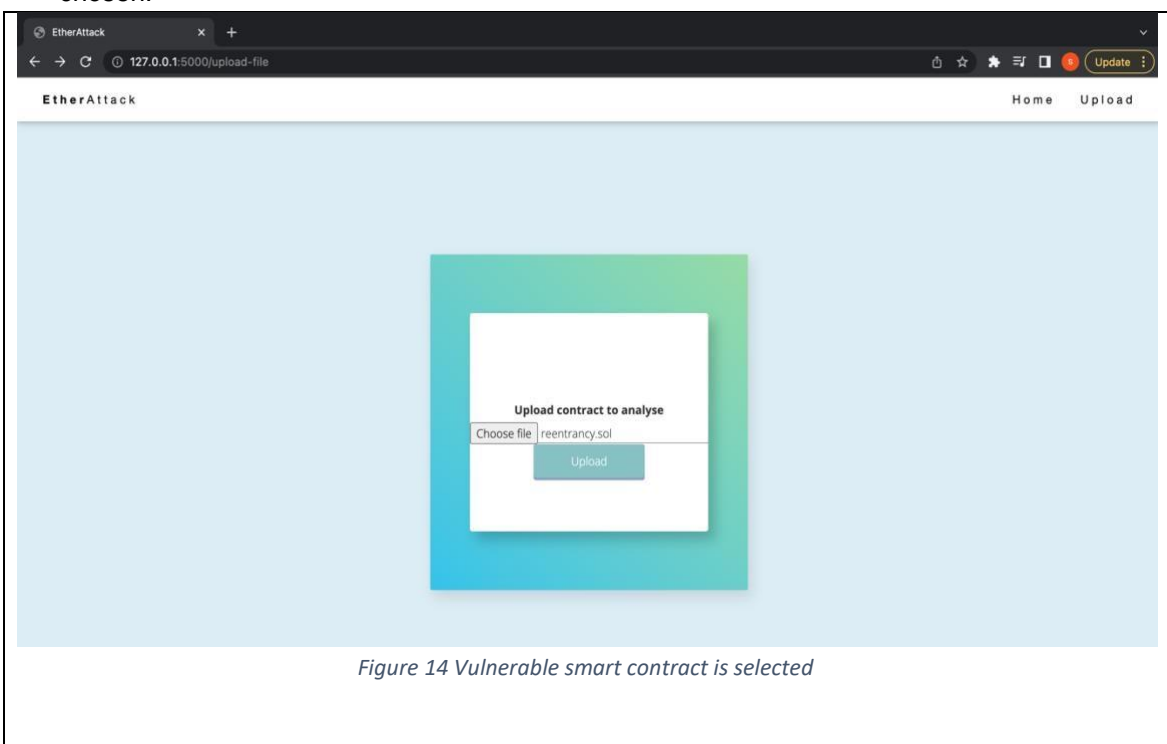


Figure 12 EtherAttack main page

- 15) Click on upload on the main page and will be directed to the upload page to upload smart contracts.



- 16) Choose a smart contract to upload. For this example a vulnerable smart contract is been chosen.



- 17) Press upload to be directed to the analysis page. Analysis page will show detail analysis. This includes Attack contract, Slither static analysis, variable data dependencies, summary of functions and contract summary.

The screenshot shows the EtherAttack web application interface. The browser address bar displays "127.0.0.1:5000/content.html/reentrancy.sol". The page has a navigation bar with "EtherAttack", "Home", and "Upload" links. The main content area is divided into four panels:

- Original Contract**: Contains Solidity code for an `EtherStore` contract with a `deposit()` and `withdraw()` function. A "Download" button is present.
- Attack Contract**: Contains Solidity code for an `Attack` contract that inherits from `EtherStore` and overrides the `withdraw()` function to call `deposit()` before withdrawing. A "Download" button is present.
- Slither Static Analysis**: Displays the results of a Slither static analysis, identifying a reentrancy vulnerability in the `withdraw()` function. It includes external calls, state variables, and a reference to the Slither documentation. A "Download" button is present.
- Variable Data Dependencies**: Shows a dependency graph for the `EtherStore` contract, illustrating the relationships between variables and their dependencies. A "Download" button is present.

Figure 15 example of the analysis page

18) Press download to download the analyzed data.

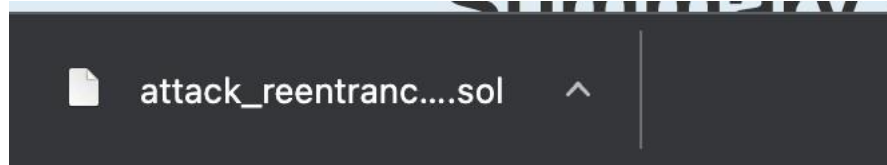


Figure 16 The downloaded file

19) If there is no vulnerable in the smart contract, a no vulnerabilities notice will be shown under the Attack Contract section in the analysis page.

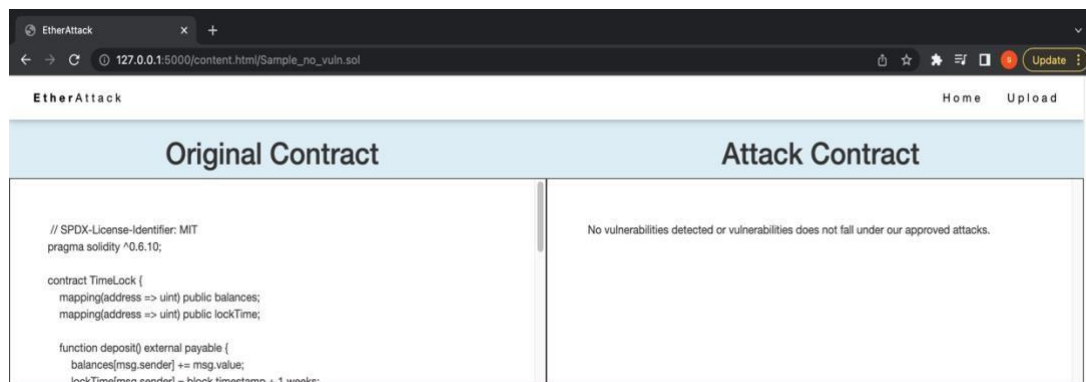


Figure 17 No vulnerable detected notice