Pythia: Identifying Dangerous Data-flows in Django-based Applications

Pythia is a static analysis tool developed by GRNET. It analyzes Django-based applications to identify well-known application vulnerabilities such as Cross-site Scripting (XSS) and Cross-site Request Forgery (CSRF).

Design and implementation details can be found in the corresponding paper found in the Proceedings of the 12th Workshop on Systems Security (EuroSec '19).

For more information about the motivation and the tool's design decisions, look here this

Features

1. Ability to parse django templates in order to find XSS vulnerabilities
2. Also tracks data from views to the templates
3. Resolves URLs to views so that we have actionable information when conducting security assessments
4. Finds *Cross Site Request Forgery* issues

Install

```
pip install django-pythia
```

How to Use

1. Setup your application's environment so that you are able to run `python manage.py runserver`
2. Install `pythia` as shown above
3. export `DJANGO_SETTINGS_MODULE` to your django's settings, e.g. `export DJANGO_SETTINGS_MODULE=myproject.settings`
4. Under your project's root, run `"export PYTHONPATH=$PYTHONPATH:${PWD}"`
5. Run `pythia`

Usage

```
usage: pythia [-h] [-i IGNORE_VARIABLES [IGNORE_VARIABLES ...]]
              [-f DANGEROUS_FILTERS [DANGEROUS_FILTERS ...]]
              [-dd DANGEROUS_DECORATORS [DANGEROUS_DECORATORS ...]] [-w] [-d]

optional arguments:
  -h, --help            show this help message and exit
  -i IGNORE_VARIABLES [IGNORE_VARIABLES ...], --ignore-variables IGNORE_VARIABLES [IGNORE_VA
                        ignore variables that appear in the output and the
                        data source is safe
```

```
  -f DANGEROUS_FILTERS [DANGEROUS_FILTERS ...], --dangerous-filters DANGEROUS_FILTERS [DANGE
                        Django filters to look for. Defaults to ["safe",
                        "safeseq"]
  -dd DANGEROUS_DECORATORS [DANGEROUS_DECORATORS ...], --dangerous-decorators DANGEROUS_DECO
                        view decorators to look for. Defaults to
                        ["csrf_exempt"]
  -w, --enable-warnings
  -d, --debug
```

By default `Pythia` looks for occurences of `safe/safeseq` filters, `@csrf_exempt` occurences and `mark_safe` invocations in view functions.