ZAP Automated Security Test

This page demonstrates a project which shows how a dev team can run ZAP headless to run automated security tests and send the results to a bug-tracker (currently only JIRA).

ZAP is an Open Source Web App Security Testing Tool and browser proxy, that is very flexible and can be automated to run as part of a build.

Project Setup

1. Download and Install ZAP - https://github.com/zaproxy/zaproxy/wiki/Downloads
2. Add ZAP root certificate to your browser - *Open ZAP* > *Tools* > *Options* > *Dynamic SSL Certificates* > *Save*
3. Configure your machine to use ZAP local proxy for all internet traffic
4. Download the project
5. Install requirements: `pip install -r requirements.txt`
6. Modify any `core/setup_module/proxy_scripts/*` as needed – all files in this folder will be used
    1. For instance, you might want to add a CSP header to each response
7. Start ZAP daemon (also see `start-zap.sh` script):
   ```
   zap.sh -daemon -port 8080 -config api.disablekey=true &
   ```
8. Setup your own selenium drivers and tests (or any other way you want to push the internet traffic)

The project is split into two parts:

run_session_setup.py is used to clean the ZAP session and set up basic configuration - this should be run before any scans are run (+ it assumes that ZAP daemon is already running)

run_scan.py contains the actual scan functions and also posts the scan results to JIRA - it assumes that ZAP daemon is running, session has been set up and selenium tests have been run (through the ZAP proxy)

Usage (relevant to both modules)

1. Run `python scriptname.py -h` to see instructions and all available options (substitute 'run_scan.py' or 'run_session_setup.py')
2. Run `python scriptname.py -g rules_config.txt` to generate a template for your rule configuration file
3. Change your `rules_config.txt` file to indicate which rules should be ignored and which should cause the test to fail
4. Modify `core/config.py` settings

Run setup_module:

1. To set up the session run:
   ```
   python run_session_setup.py -t "www.example.com" -c rules_config.txt -d
   ```

Run scan_module:

1. Run: `python run_scan.py -c rules_config.txt -r` to execute the test and have the results posted to JIRA (Note: You should use the same rules_config.txt file!)