5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopted test strategy and instruments. In this overarching introduction, given an overview of the testing strategy and your team's overall testing philosophy. Emphasize any unique challenges to testing for your system/design.

In the sections below, describe specific methods for testing. You may include additional types of testing, if applicable to your design. If a particular type of testing is not applicable to your project, you must justify why you are not including it.

When writing your testing planning consider a few guidelines:

- Is our testing plan unique to our project? (It should be)
- Are you testing related to all requirements? For requirements you're not testing (e.g., cost related requirements) can you justify their exclusion?
- Is your testing plan comprehensive?
- When should you be testing? (In most cases, it's early and often, not at the end of the project)

5.1 Unit Testing

What units are being tested? How? Tools?

We have several different units of our project that are being tested as they are implemented. The interface used for our grid is one of the units that is tested to ensure that the parameters used are working properly and working functionally as expected. We also unit test the simulated grid by checking to make sure that the grid converges and is simulated and complies correctly. For these testing purposes we will be utilizing PandaPowers built-in compiler and any additional Python library we may need to make a functionally simulated power grid. Using Python 3.9 we will also be creating the scripts used for the attack vectors of the project. This leads into another unit of the project that will be tested; by executing the scripts alongside the simulation of the grid we will need to verify that the expected results match those of the results received.

5.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

The only interfaces that we use are the PandaPower electricity grid that we generate through Python and then PyCharm as a compiler. We could also use online GitHub, but it won't be necessary for testing. As we test PyCharm, we also test PandaPower as both are synonymously tested. This can be done with a simple script and as long as we have a valid interpreter, it should work if nothing was corrupted in the install process.

5.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

The critical integration paths in our design are verifying if instances of attacks work, and their correctness. Verifying these tests will include running an attack script on our grid and then simulating it in PandaPower. Following this if the grid does not converge we will need to verify a blackout occurred. Otherwise we will need to verify the attack was successful and the outputs are reasonable. The tools used for this will be our attack scripts, PandaPower, and our grid.py.

5.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

The essential parts for testing the system as a whole is ultimately making sure that the fundamental components are working properly. Since the only components we use is Python, PandaPower, and GitHub, this makes things easy. For Python, we just need a simple test program that uses the libraries we will need to verify they work. PandaPower is a library in Python, so this can be tested in conjunction with Python. GitHub is also quite easy to test as we can do simple push and pull requests to see if our code uploads and downloads respectively.

5.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?

We will confirm that new features don’t break old ones by continuously testing as soon as a new feature is implemented. We will utilize Panda Power’s simulation environment and different wPython compilers to verify any warnings that come up, and also create use-case tests to ensure that using a new feature doesn’t brick an old one, and vice versa, just to make clear that the features are preserved in new updates. We can set up these tests in GitHub by using a pipeline, or by making a build-deploy script that will be run before a commit/pull request.

5.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

In order to show that the requirements have been met for both functional and non-functional requirements users will test the project at several stages. The users will consist of employees who would work with the program. We want to ensure that the interface functions as intended, the interface has a comprehensible use-case statement (--help, -h), the interface easy to understand and use, and grid.py was generated. The users will complete testing by; running the interface, running any commands to create their specific grid, and test if the user has questions, use-case statements will help.

5.7 Security Testing (if applicable)

Not applicable

5.8 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

The results of these conducted tests will help to further improve and enhance our procedure for completing the project and design. The results ensure compliance through proving and providing visualizations of the outcomes so individuals are able to physically see the different outcomes from the attack vectors. Once the results are created for the designated user, they will then be able to implement any countermeasures as desired. Our intended use for the final product is for companies to be able to take the information that is simulated from our project and use that to become more aware of ongoing threats.