AT-AT – Testing Document

Group 8

Anandarajah Yathuvaran

Anil Menon

Dylan Leveille

Eric Leung

Supervisor: Dr. Jason Jaskolka

SYSC 4907 Engineering Project

Department of Systems and Computer Engineering
Faculty of Engineering and Design Carleton University

Last Updated: February 25, 2022

Table Of Contents

Introduction

The purpose of this document is to provide test scenarios for the AT-AT tool. Due to the tool being a web application, usual automated testing tools are not suitable to test the application. As such, this document was created to test unique features in the application and help future developers in ensuring the application is still functional.

This testing document follows some of the standards found in IEEE 829, but with a few modifications to make it more concise.

Features to be Tested

The features to be tested are focused on basic functionalities in the software, and some of the edge cases a user may perform in the system. All tested features are things a user can see. These features are based on the use cases described in the SRS document. The list of features to be tested are:

- Input DSL's
- Imported DSL's
- Exported DSL's
- Switching Tabs
- Tree Generation
- Attack Scenarios
- Recommendations

Features not to be Tested

The features not to be tested include things that a user would not be able to see (i.e., backend logic is not rigorously tested). The list of non-tested features is:

- Tree analysis algorithm
- Tree storage
- Internal program state

Approach

The technique and type of testing we are going to use is state-based testing. No special tools are to be used and no special training is required. No metrics are to be collected for testing. Configuration management is not handled, and a singular release configuration will be tested. As the application is platform-independent; it can be tested the same on Windows, macOS, and major flavors of Linux. No levels of regression testing are done. The overall project testing approach is state-based testing and there are no coverage requirements. There are no specific requirements for the testing.

Test Cases

1.0 Import DSL

Initial State:

1. Application is open.
2. No text in the DSL input box.

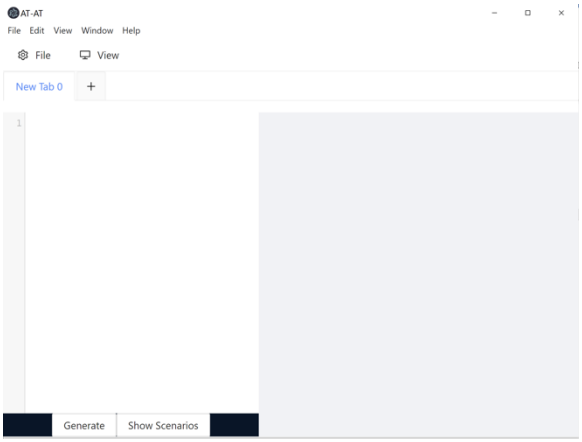


Figure 1: Empty DSL input box

Steps:

Step 1: Click “File”

Step 2: Click “Import DSL”

Step 3: Select Text File containing DSL within file explorer

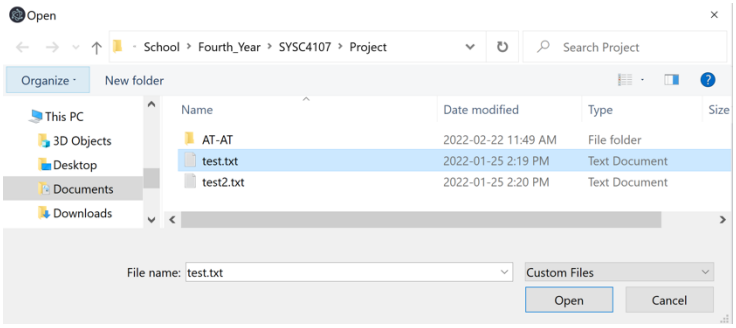


Figure 2: File explorer

Step 4: Click Open

Final State:

The content in the text file is displayed

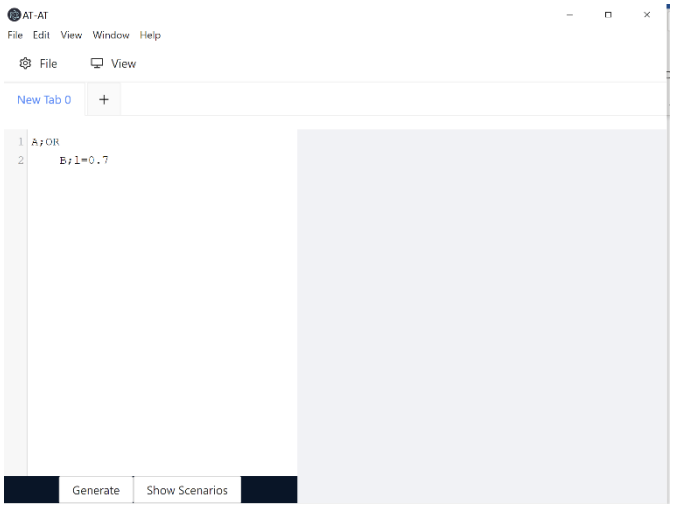


Figure 3: DSL input box now includes the content of text file imported.

2.0 Import – File Format Not .txt

Initial State:

1. Application is open.
2. No text in DSL input.

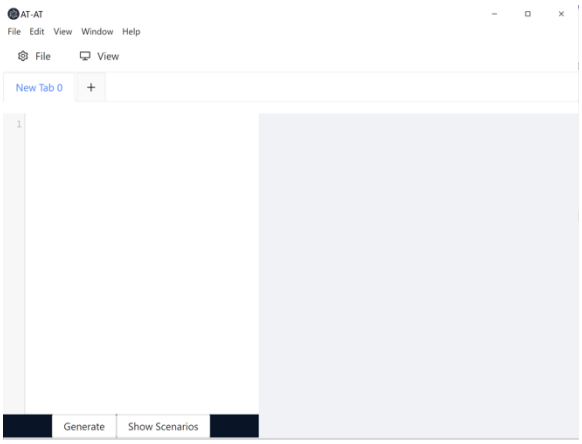


Figure 4: Empty DSL input box

Steps:

Step 1: Click “File”

Step 2: Click “Import DSL”

Step 3: Select File that isn’t a text file

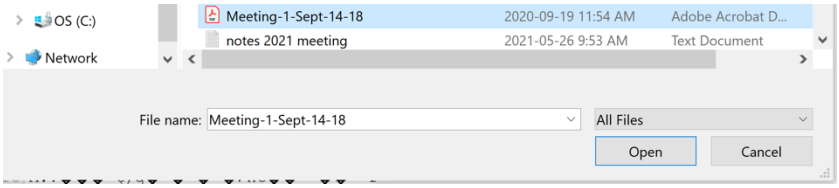


Figure 5: File explorer with type that is not .txt

Step 4: Click Open

Final State:

The content in the file is displayed as text in the text box

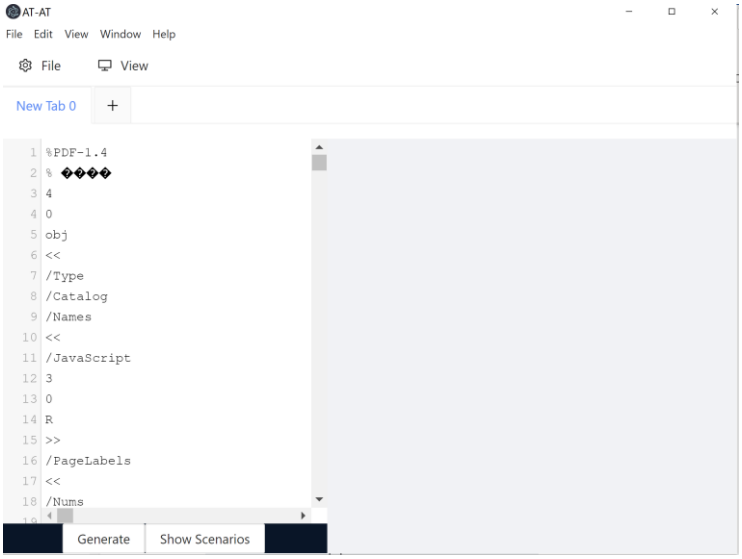


Figure 6: Incorrect DSL syntax is imported into DSL input box.

3.0 Export DSL

Initial State:

1. Application is open.
2. DSL formatted text as presented below is in DSL input box.

```
a;OR
        b;l=0.7;v=0.1;t=0.2;r=0.5
        c
        d
        e;AND
            f
            g
            h
```

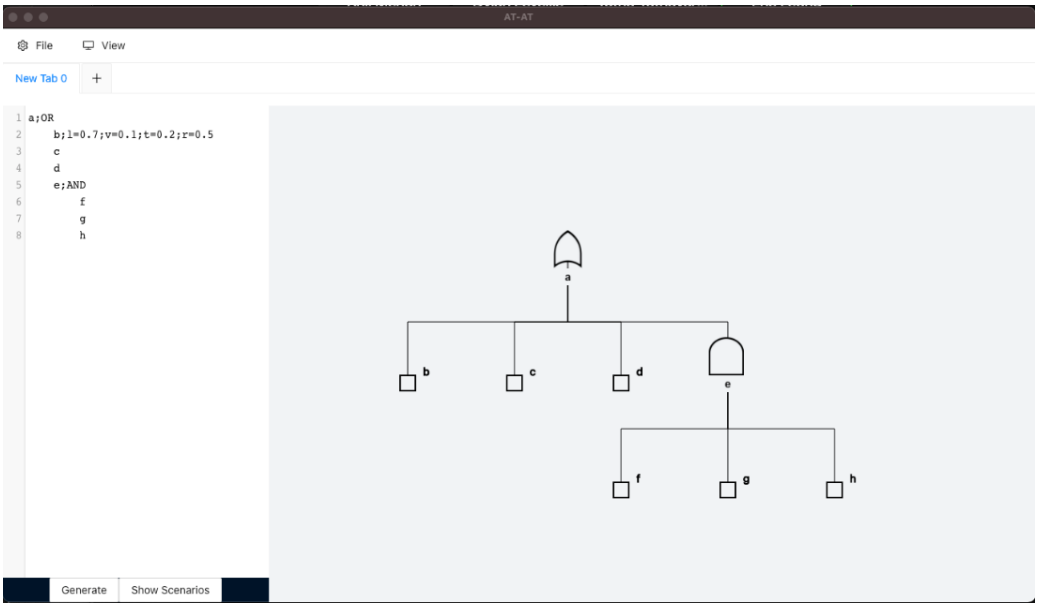


Figure 7: Tree generated using the DSL entered in DSL input box.

Steps:

Step 1: Click “File”

Step 2: Click “Export DSL”

Step 3: Choose destination folder

Step 4: Filename name must end with .txt

Step 5: Click “Save”

Final State:

DSL is exported as a .txt file in your destination folder

Figure 8: Text file found in destination folder

The contents of the DSL file should be the same as the inputted DSL text

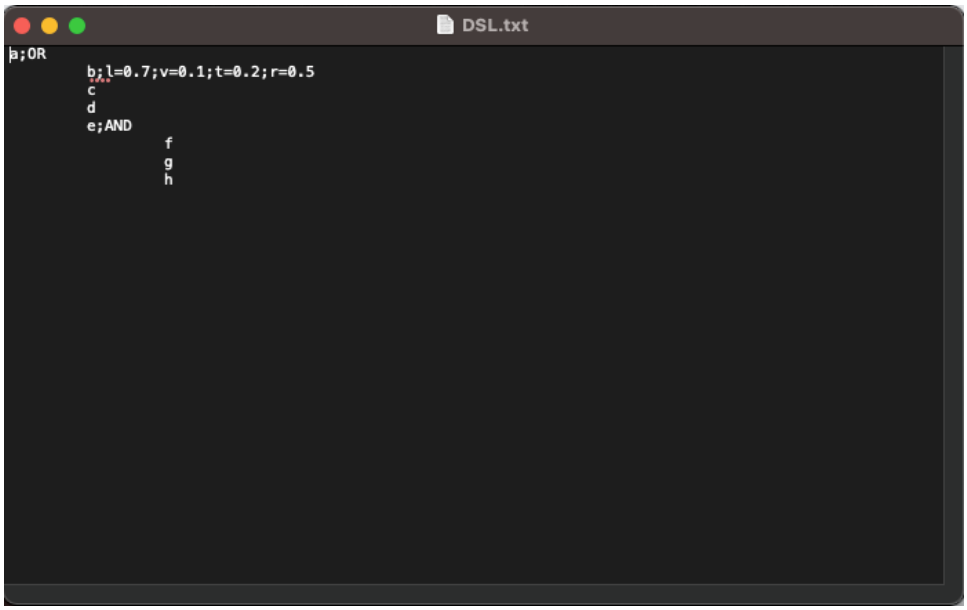


Figure 9: Contents exported properly into DSL.txt file

4.0 DSL – Improper Tabbing

Initial State:

1. Application is open.
2. DSL formatted text as presented below is in DSL input box.

```
a;OR
        b;l=0.7;v=0.1;t=0.2;r=0.5
            c
            d
            e;AND
                f
                g
                h
```

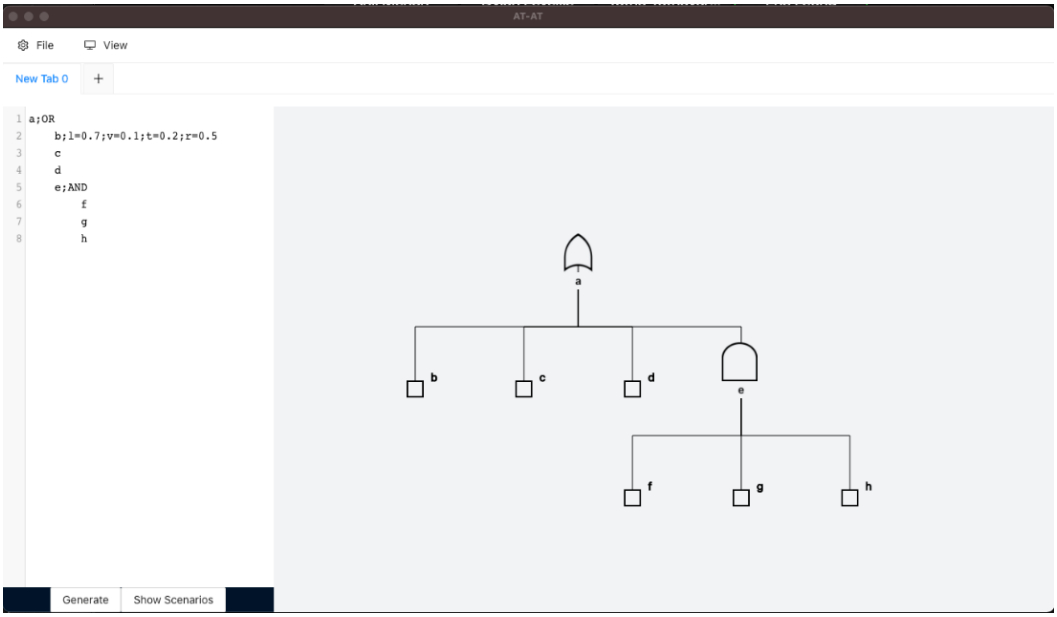


Figure 10: DSL added into DSL input box and tree generated

Steps:

Step 1: Add a tab to line 2

Step 2: Click “Generate”

Final State:

The application will provide an error message on the top right corner indicating there is a tabbing error on line 2.

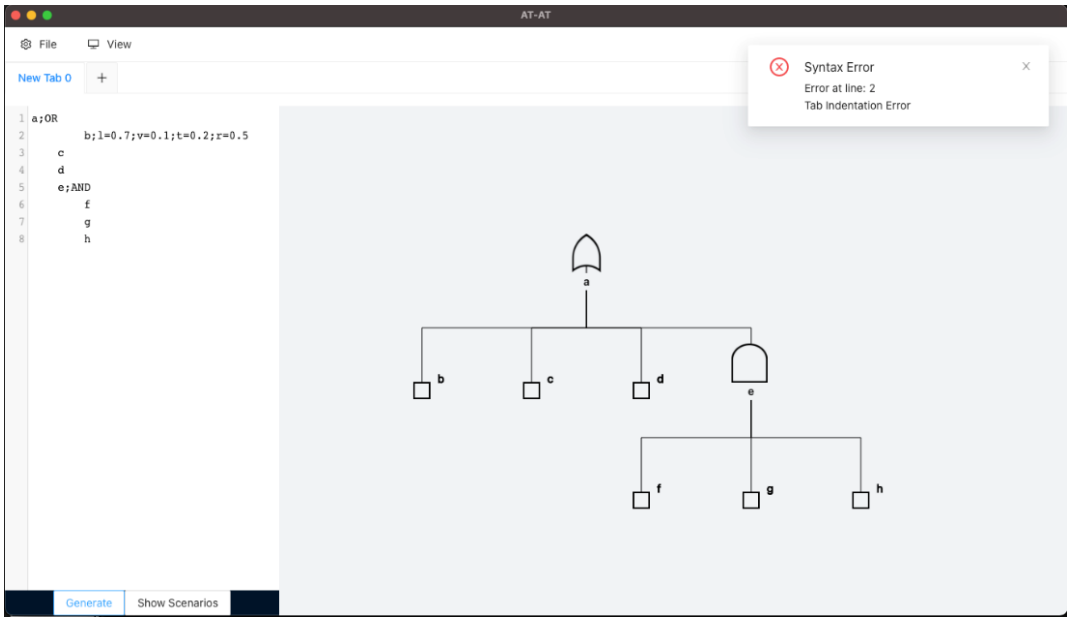


Figure 11: Tabbing error notification

5.0 DSL – Improper Parent Node Syntax – Invalid Logic Gate

Initial State:

1. Application is open.
2. DSL below is in box.

```
openSafe;AND
        plantDynamite
        IgniteDynamite
```

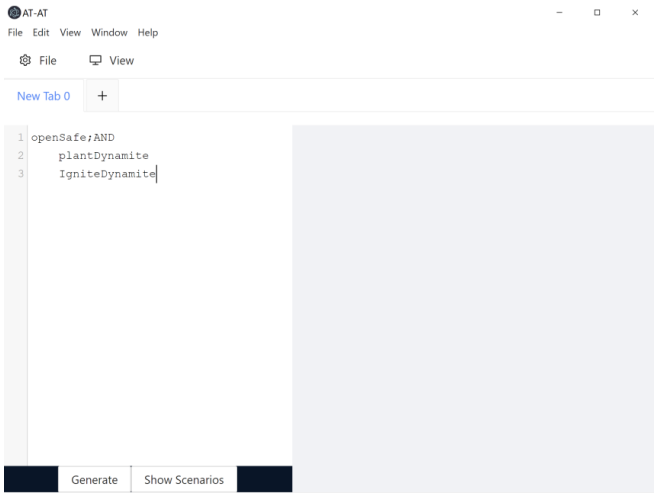


Figure 12: DSL added into DSL input box and tree generated

Steps:

Step 1: Change the AND node to an invalid gate, e.g., ANDD

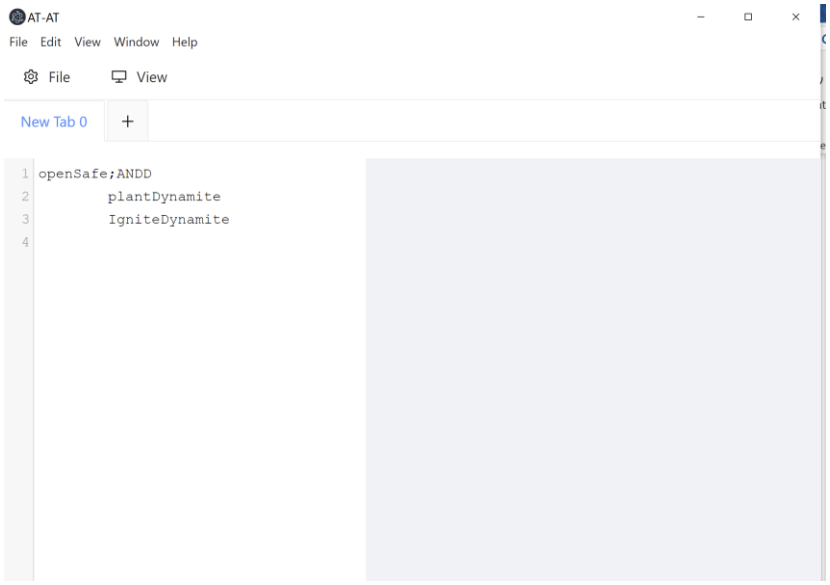


Figure 13: “AND” changed to “ANDD” in inputted DSL

Step 2: Click Generate

Final State:

The application will provide an error message on the top right corner indicating there is a syntax error.

Figure 14: Syntax error notification

6.0 DSL – Improper Parent Node Syntax – Invalid Node Name

Initial State:

1. Application is open.
2. DSL below is in box.

```
openSafe;AND
        plantDynamite
        IgniteDynamite
```

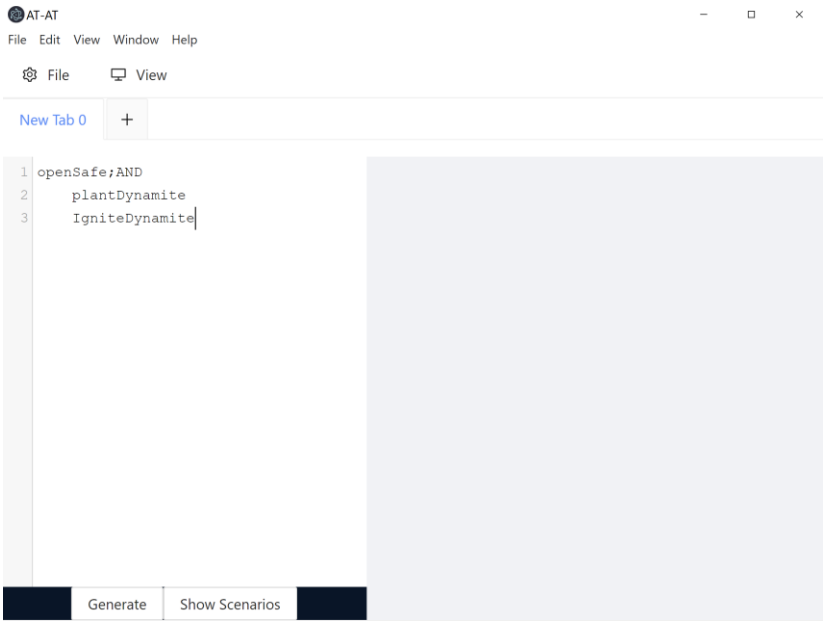


Figure 15: DSL added into DSL input box and tree generated

Steps:

Step 1: Change the node to an invalid gate, e.g., blank

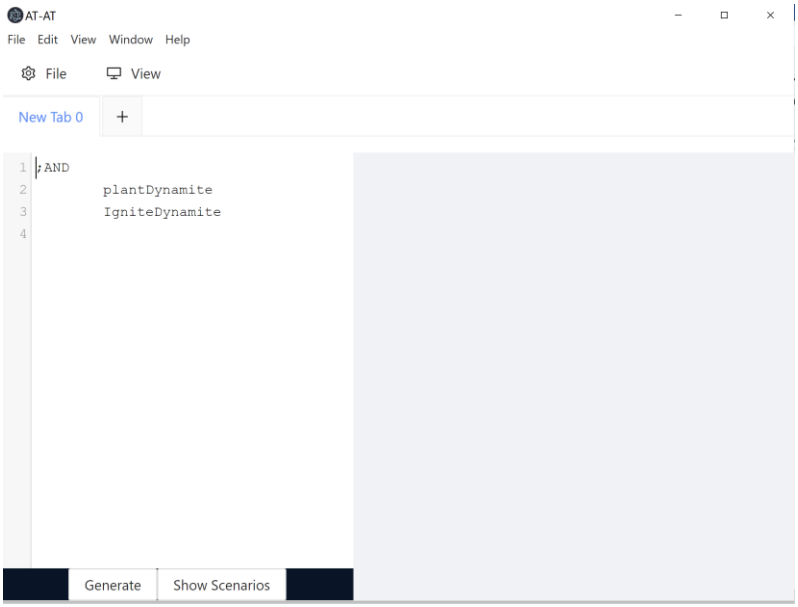


Figure 16: Removed node name from line 1 and added blank

Step 2: Click Generate

Final State:

The application will provide an error message on the top right corner indicating there is a syntax error.

Figure 17: Syntax error notification

7.0 DSL – Improper Leaf Node Syntax – Improper Leaf Name

Initial State:

1. Application is open.
2. DSL formatted text as presented below is in DSL input box.

```
openSafe;AND
        plantDynamite
        IgniteDynamite
```

Steps:

Step 1: Add a tab to the leaf node named “dynamite” anywhere within the name. E.g.: “dyna mite”.

Final State:

The application will provide an error message on the top right corner indicating there is a syntax error.

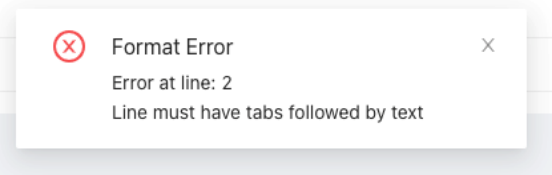


Figure 18: Syntax error notification

8.0 DSL – Improper Leaf Node Syntax – Metric Values Too Big

Initial State:

1. Application is open.
2. DSL below is in box.

openSafe;AND

plantDynamite;l=0.5

IgniteDynamite;l=0.9;v=0.6

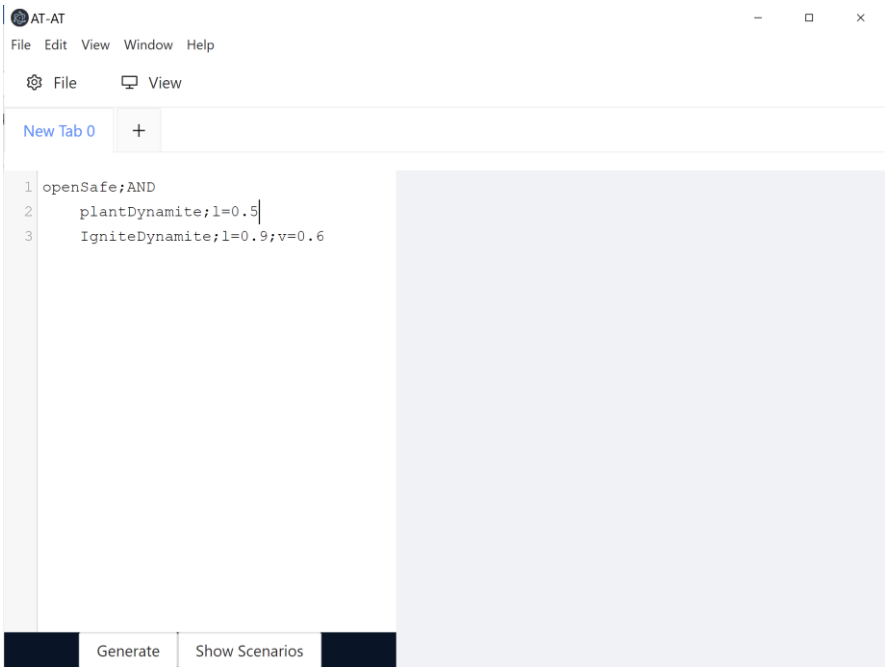


Figure 19: DSL added into DSL input box and tree generated

Steps:

Step 1: Change a node's metric to value above 1

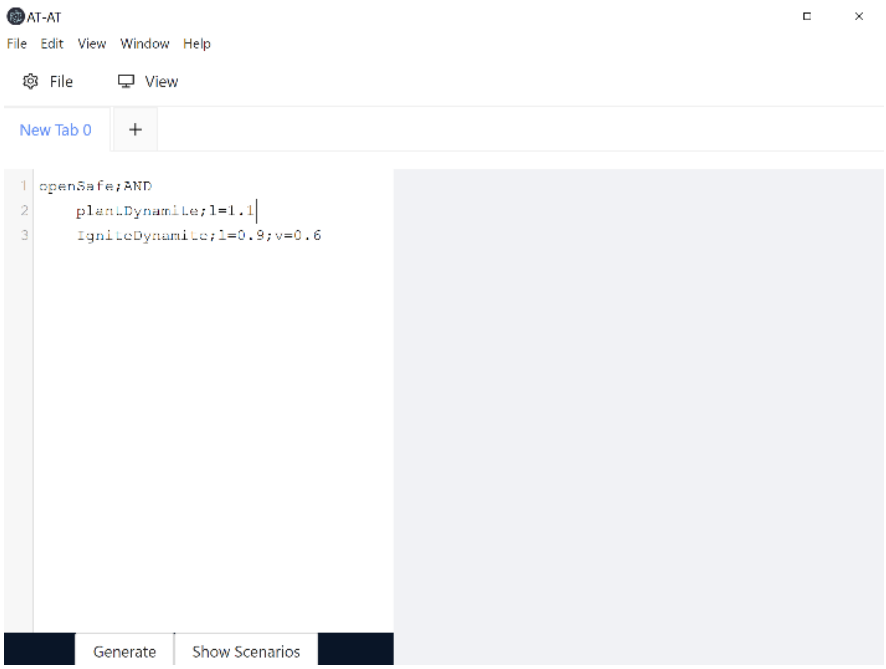


Figure 20: Metric value of node in line 2 changed to value above 1

Step 2: Click Generate

Final State:

The application will provide an error message on the top right corner indicating there is a syntax error in the leaf node.

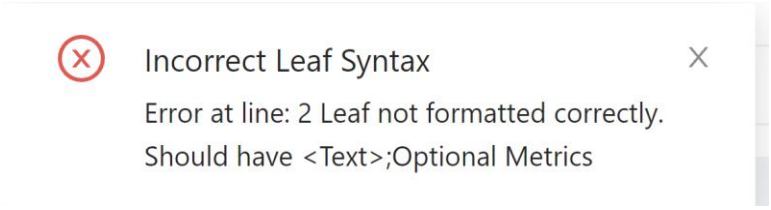


Figure 21: Leaf syntax error notification

9.0 Tree Generation – Wide Tree

Initial State:

1. Application is open.
2. DSL below is in box.

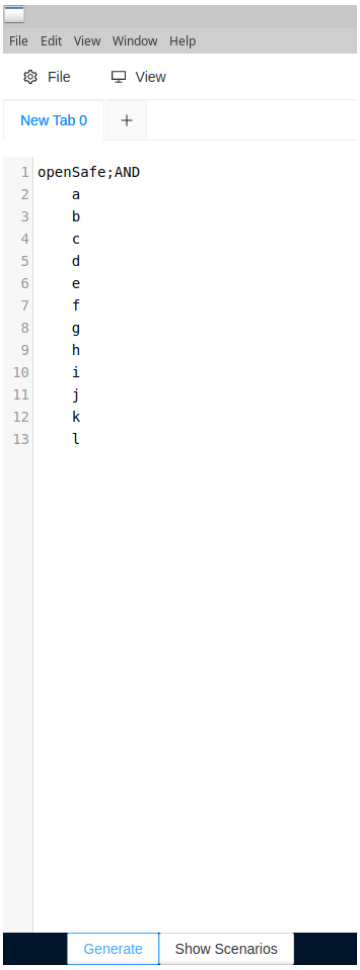


Figure 22: DSL input box

Steps:

Step 1: Click Generate

Final State:

The application will display the tree correctly with a node cut off. The entire tree can be seen by panning or zooming out.

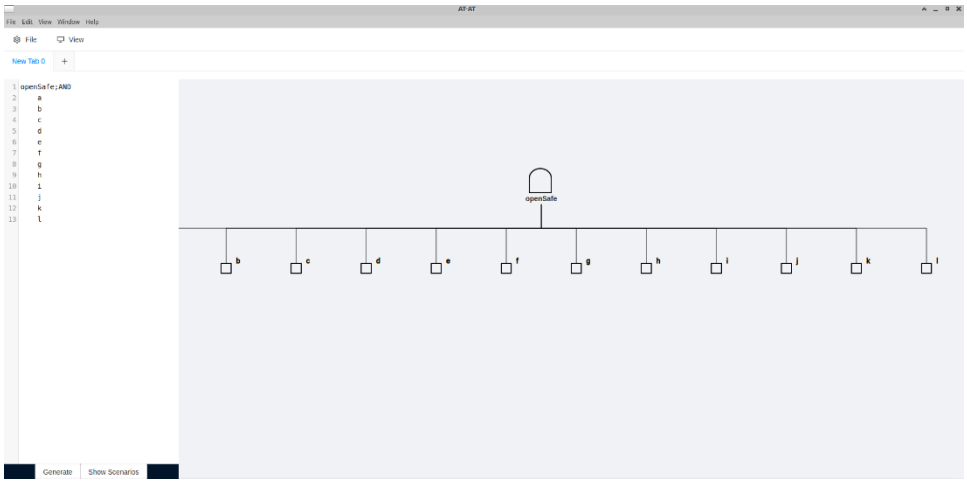


Figure 23: A wide tree generated based on DSL input

Note that upon zooming out, all nodes are present.

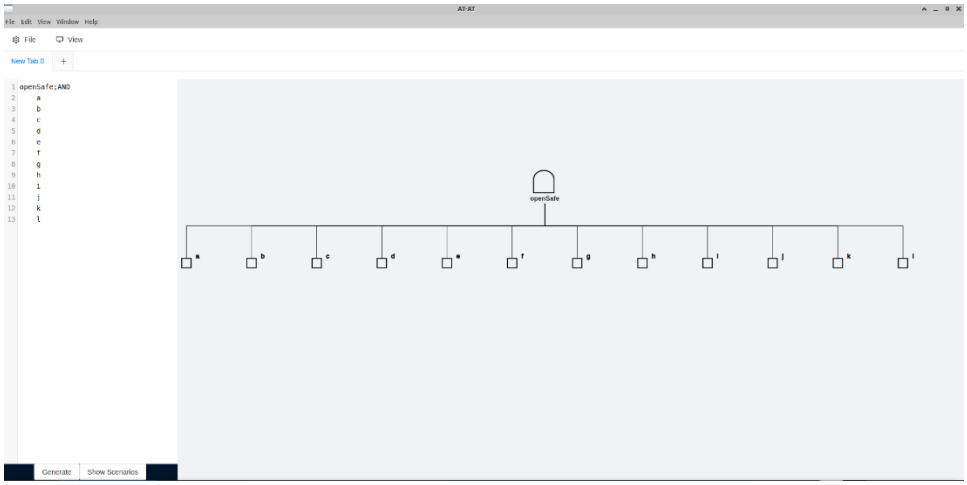


Figure 24: Zooming capability of tree box

10.0 Tree Generation – Long Tree

Initial State:

1. Application is open
2. DSL below is in box.

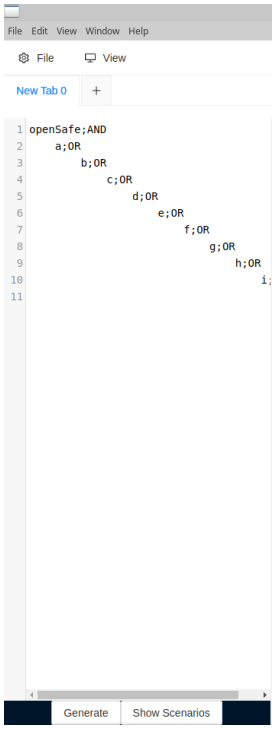


Figure 25: DSL input box left scroll

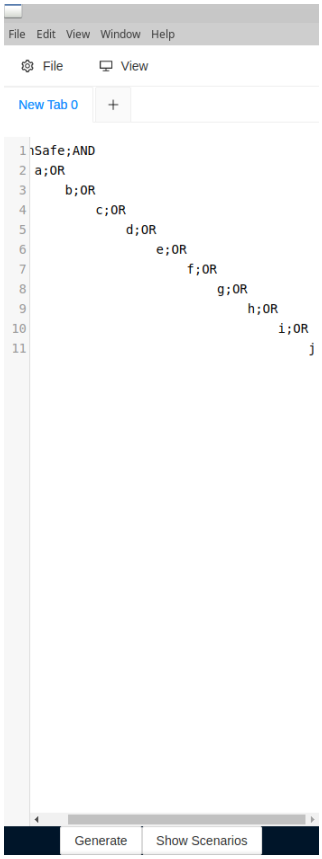


Figure 26: DSL input box right scroll

Steps:

Step 1: Click Generate

Final State:

The application will display the tree correctly with nodes cut off at the bottom. The entire tree can be seen by panning or zooming out. There should also be a horizontal scroll bar for the text area.

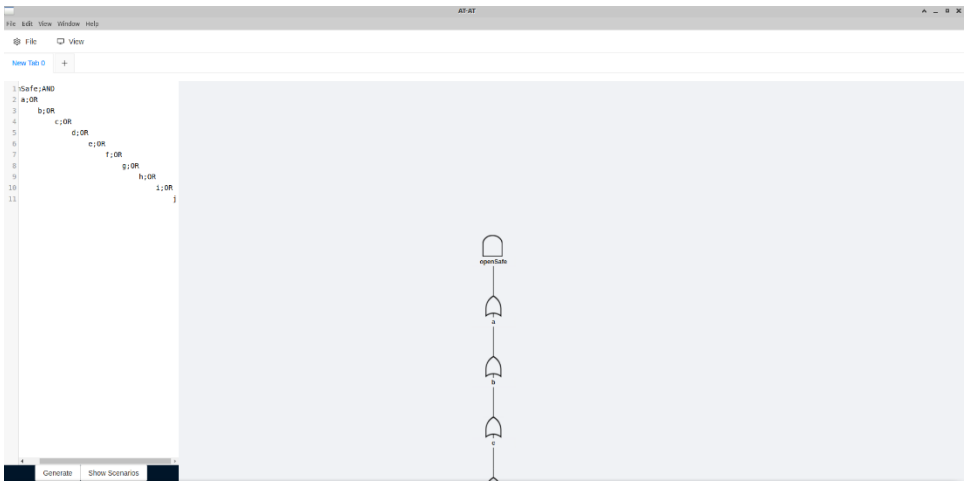


Figure 27: Tree visibility without zooming

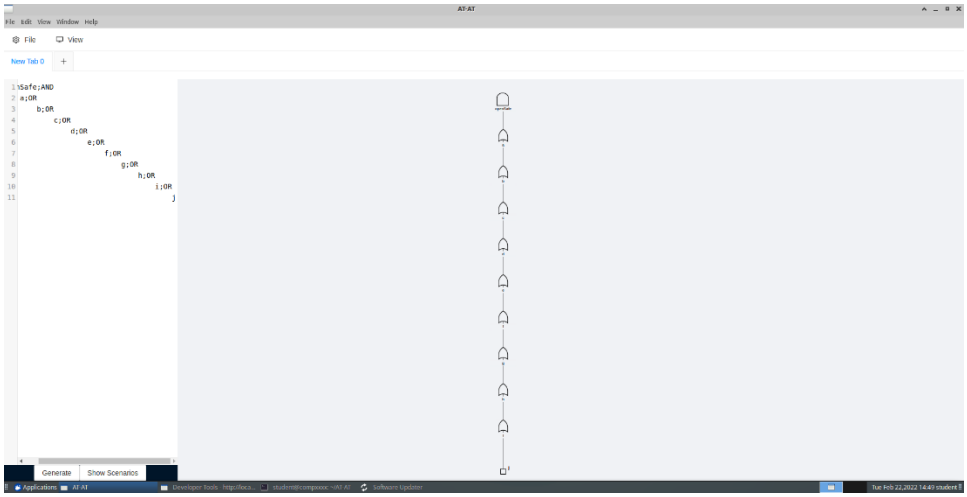


Figure 28: Tree visibility after zooming out

11.0 Tree Switching – Tree Saved Properly

Initial State:

1. Text is entered in the DSL entry box.
2. Text can be in any style/format.

Figure 29: Node name in any style/format

Steps:

Step 1: Click the “+” icon to switch to a new tab

Step 2: Return to previous tab

Final State:

Text is intact and still in the box.

Figure 30: Dummy text in DSL does not change after tab switching

12.0 Tree Switching – Tree Removes Highlighting

Initial State:

1. Tree generated using a DSL.
2. A path has been selected and highlighted.

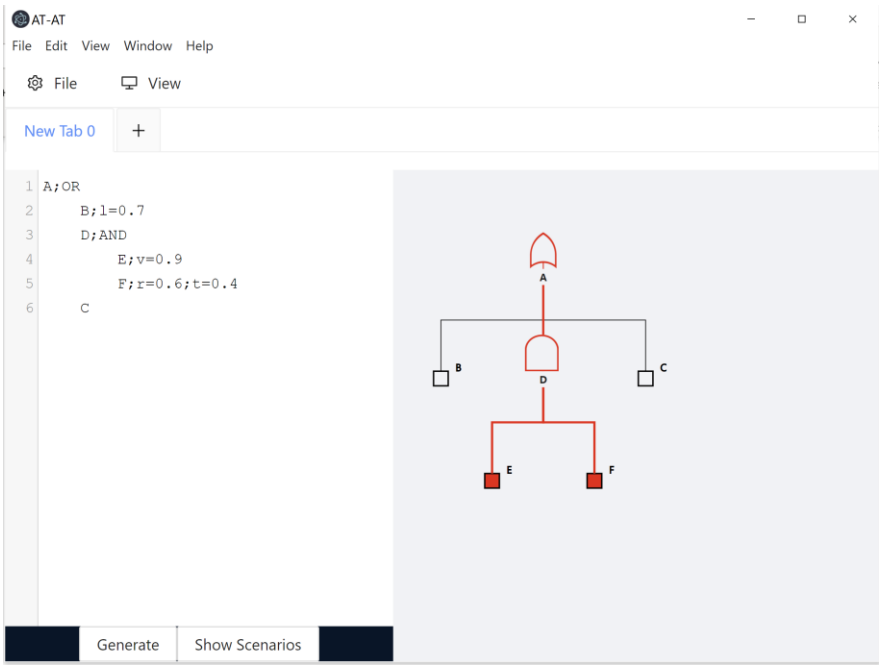


Figure 31: DSL entered, and path is selected

Steps:

Step 1: Open a new Tab by clicking the “+” button.

Step 2: Return to previous tab by clicking the tab for the previous tree

Final State:

The tree in the previous tab is not highlighted.

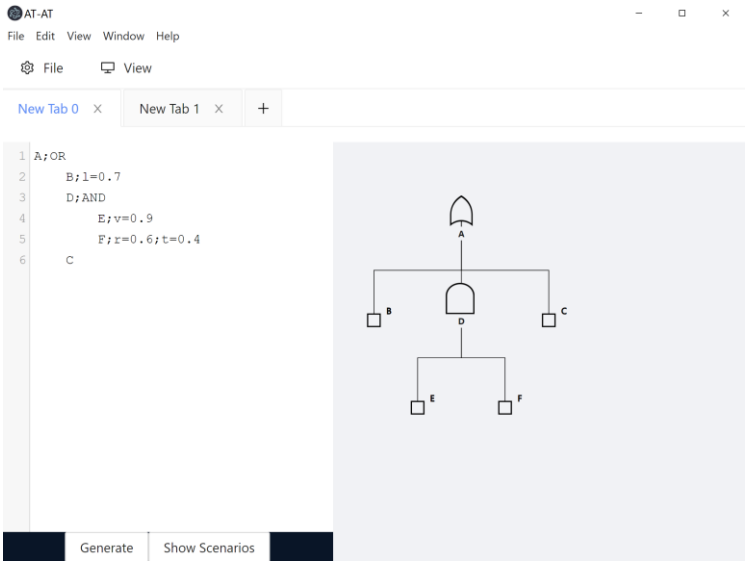


Figure 32: Tree no longer highlighted after returning

13.0 Attack Scenarios – Severity Listed

Initial State:

1. Application is open.
2. DSL below is in the box. This DSL is crafted in such a way that the paths, listed in order of severity, should be (A, D, E, F), (A, B), (A, C).

```
A;OR
        B;l=0.7
        D;AND
                E;v=0.9
                F;r=0.6;t=0.4
        C
```

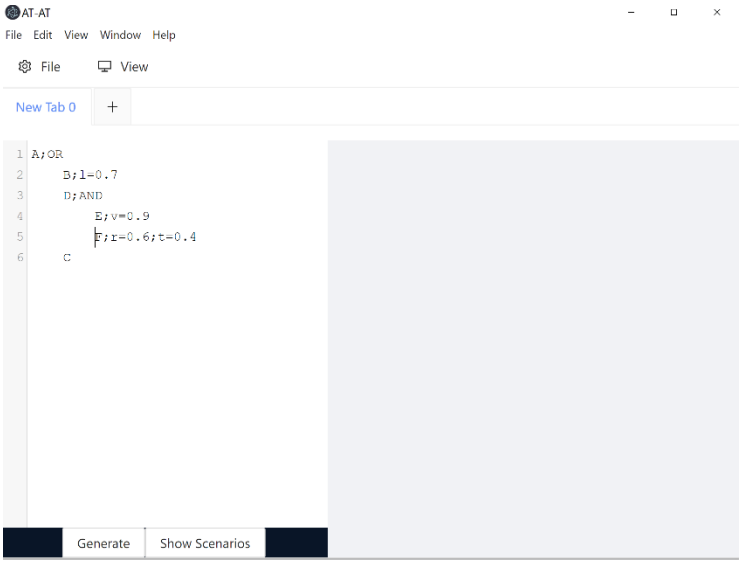


Figure 33: DSL entered in DSL box

Steps:

Step 1: Click Generate

Step 2: Click Show Scenarios

Final State:

Three scenarios with the following severities for each metric:

Scenario	Severity			
	L	V	R	T
Scenario 1	N/A	0.90	0.60	0.40
Scenario 2	0.70	N/A	N/A	N/A
Scenario 3	N/A	N/A	N/A	N/A

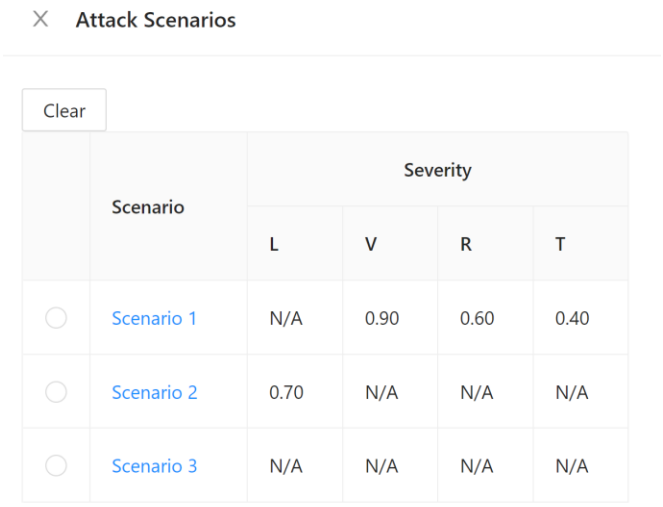


Figure 34: Attack scenarios with respective severities

14.0 Attack Scenarios – No Severity

Initial State:

1. Application is open.
2. DSL below is in box. This DSL does not include any metrics.

```
a;OR
        b
        d
        e;AND
                f
                g
                h
```

Steps:

Step 1: Click “Generate”

Step 2: Click “Show Scenarios”

Final State:

The screenshot below indicates no severities for any of the scenarios since no metrics were provided.

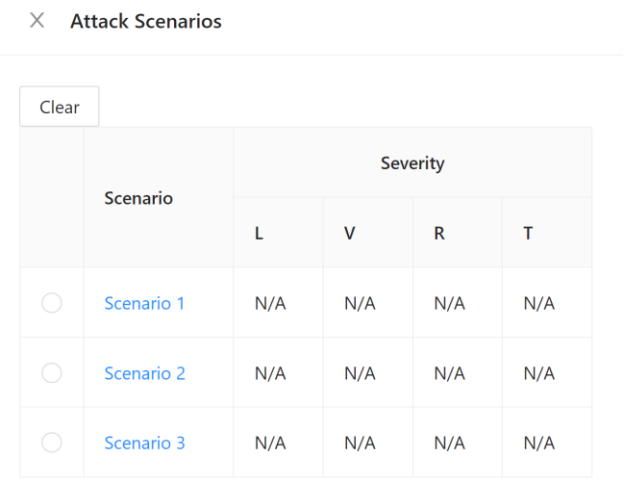


Figure 35: Scenarios with no severity

15.0 Attack Scenarios – Proper Highlighting

Initial State:

1. Application is open.
2. DSL below is in box.

```
a;OR
        b
        d
        e;AND
                f
                g
                h
```

Steps:

Step 1: Click “Generate”

Step 2: Click “Show scenarios”

Step 3: Click on Scenario 1 radio button

Final State:

The screenshot below displays scenario 1 by highlighting its path in red.

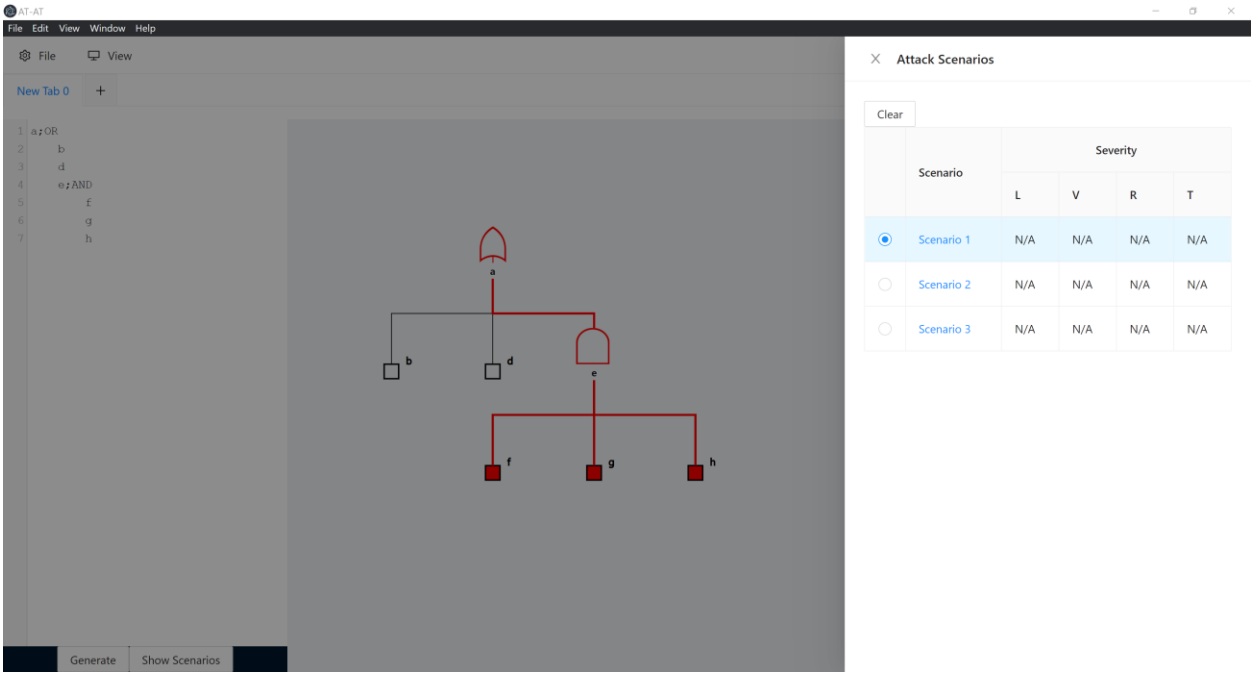


Figure 36: Scenario 1 highlighted in red

16.0 View Recommendations – Box Displays

Initial State:

1. Application is open.
2. DSL below is in box.

Steps:

Step 1: Click “Generate”

Step 2: Click “Show Scenarios”

Step 3: Click scenario 1 radio button

Step 4: Click “View”

Step 5: Click “Enable Recommendations”

Final State:

The output shown in the screenshot below indicates that a recommendations box has appeared on screen.

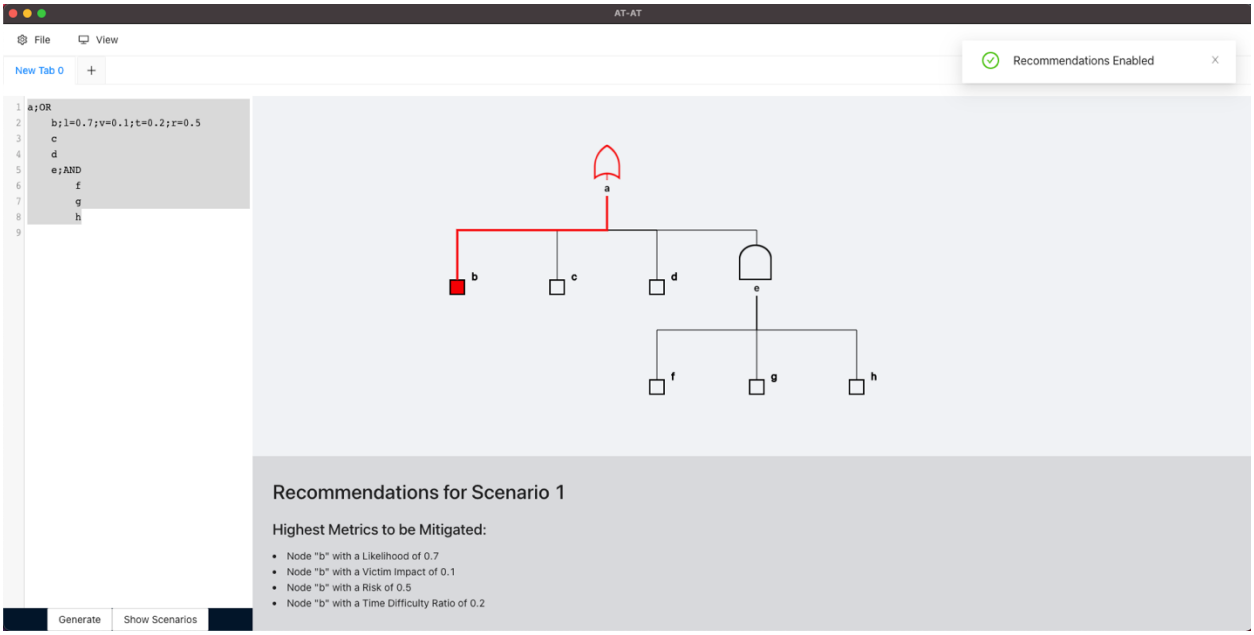


Figure 37: Recommendation’s box appears after enabling recommendations

17.0 View Recommendations – Remove Box

Initial State:

1. Application is open.
2. DSL below is in box and the tree has been generated.
3. The View Recommendations option is enabled.
4. A scenario is selected.

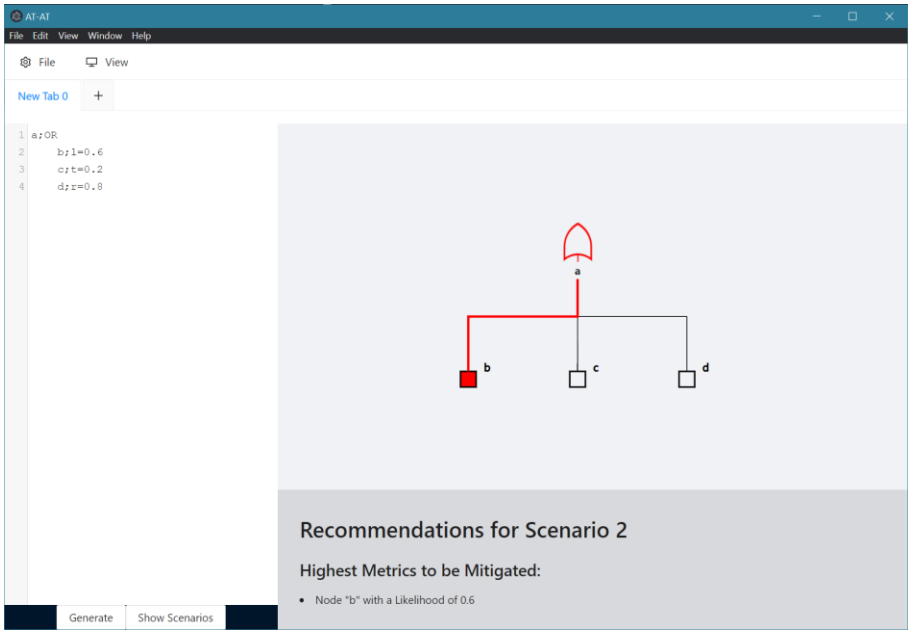


Figure 38: DSL input, tree generated, recommendations enabled.

Steps:

Step 1: Hover the mouse cursor over the “View” button on the top menu bar

Step 2: Click the “Disable Recommendations” button

Final State:

The output shown in the screenshot below indicates that the recommendations box has been disabled and hidden from view on the screen.

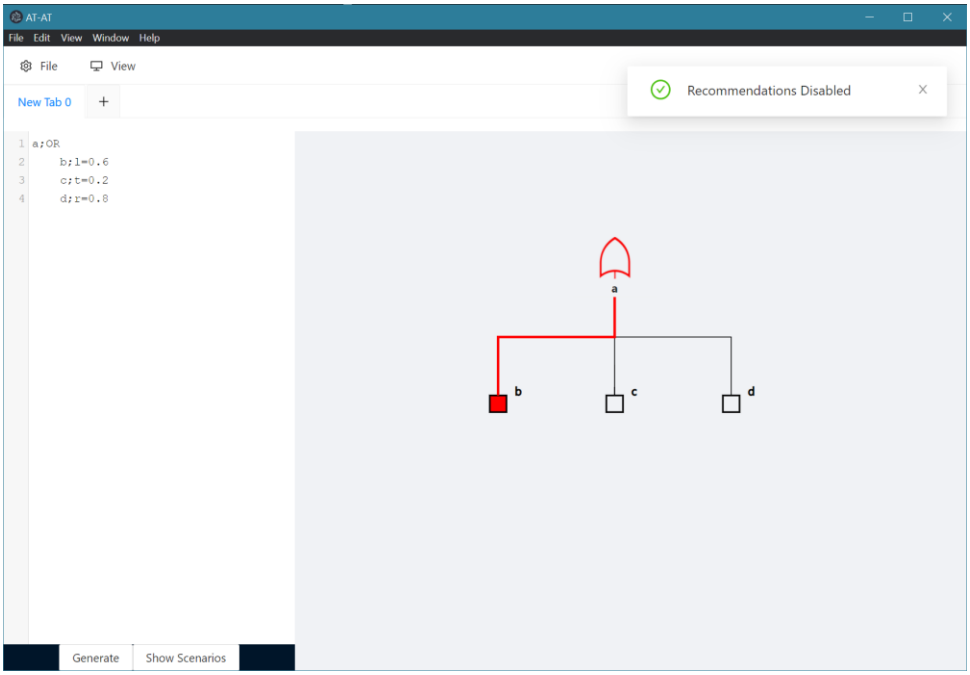


Figure 39: Recommendation's box disappears after disabling recommendations

18.0 Recommendations – Common Recommendations Display

Initial State:

1. Application is open.
2. DSL below is in box.

```
a;OR
        b;l=0.7;v=0.1;t=0.2;r=0.5
        c
        phishing
        e;AND
                f
                g
                h
```

Steps:

Step 1: Click "Generate"

Step 2: Click “Show Scenarios”

Step 3: Select scenario 3 radio button

Step 4: Select “View”

Step 5: Select “Enable Recommendations”

Final State:

The screenshot below displays scenario 3 which includes the term “phishing”. Since we have found a keyword related to cybersecurity, we have provided custom recommendations to mitigate an attack such as “phishing”.

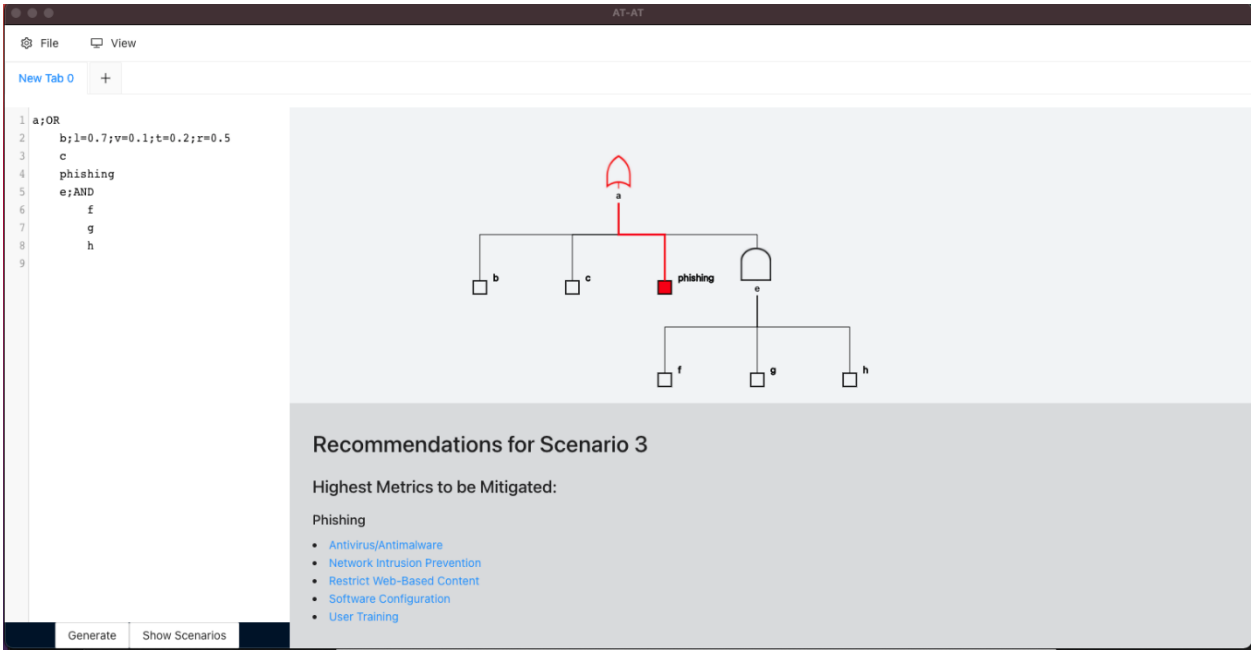


Figure 40: Recommendation box includes mitigation links for the key term “phishing”

19.0 Recommendations – Top Metrics Display

Initial State:

1. Application is open.
2. DSL below is in box.

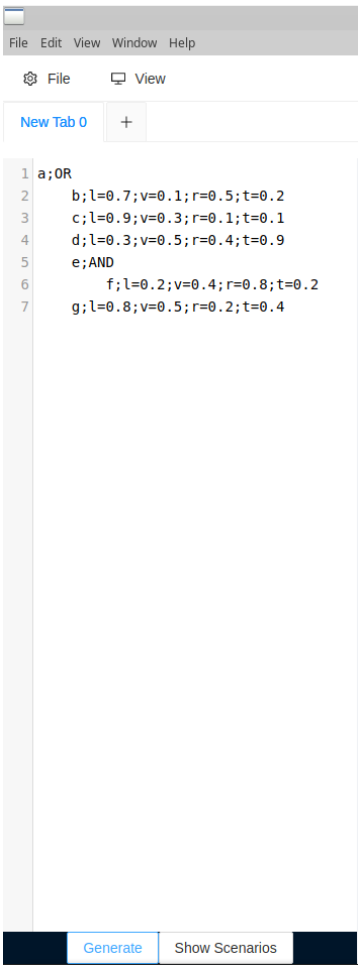


Figure 41: DSL input added to the DSL input box

Steps:

Step 1: Click “Generate”. The following tree should be generated.

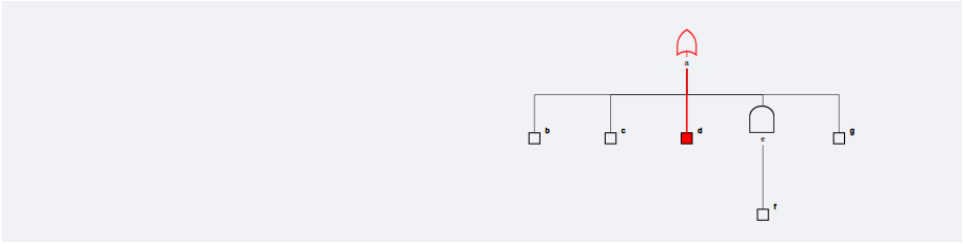


Figure 42: Tree generated based on input DSL

Step 2: Click “View”

Step 3: Click “Enable Recommendations”

Step 4: Click “Show Scenarios”

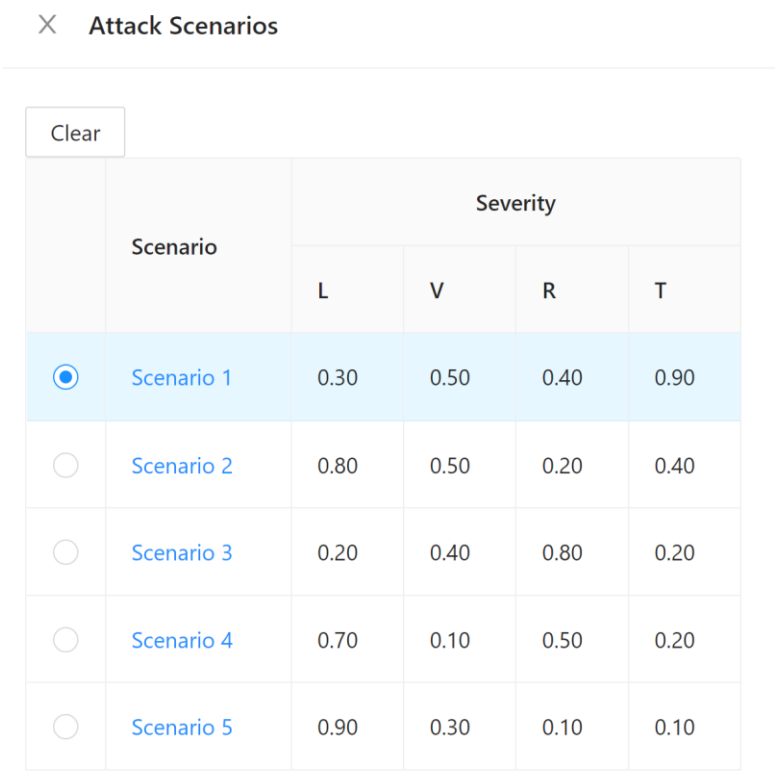


Figure 43: List of scenarios and severities listed after clicking “Show Scenarios”

Final State:

For scenario 1, the highest metrics should all be d.

Figure 44: Displays highest metrics is node d

For scenario 2, the highest metrics should all be g.

Figure 45: Displays highest metrics is node g

For scenario 3, the highest metrics should all be f.

Figure 46: Displays highest metrics is node f

For scenario 4, the highest metrics should all be b.

Figure 47: Displays highest metrics is node b

For scenario 5, the highest metrics should all be c.

Figure 48: Displays highest metrics is node c

20.0 Report Generation – Report Can Be Displayed and Saved

Initial State:

1. Application is open.
2. DSL below is in box.

a;OR

b;l=0.7;v=0.1;t=0.2;r=0.5

c;l=0.9;v=0.3;t=0.1;r=0.1

d;l=0.3;v=0.5;t=0.9;r=0.4

e;AND

f;l=0.2;v=0.4;t=0.2;r=0.8

g;l=0.8;v=0.5;t=0.4;r=0.2

Steps:

Step 1: Select “File”

Step 2: Select “Generate Report”

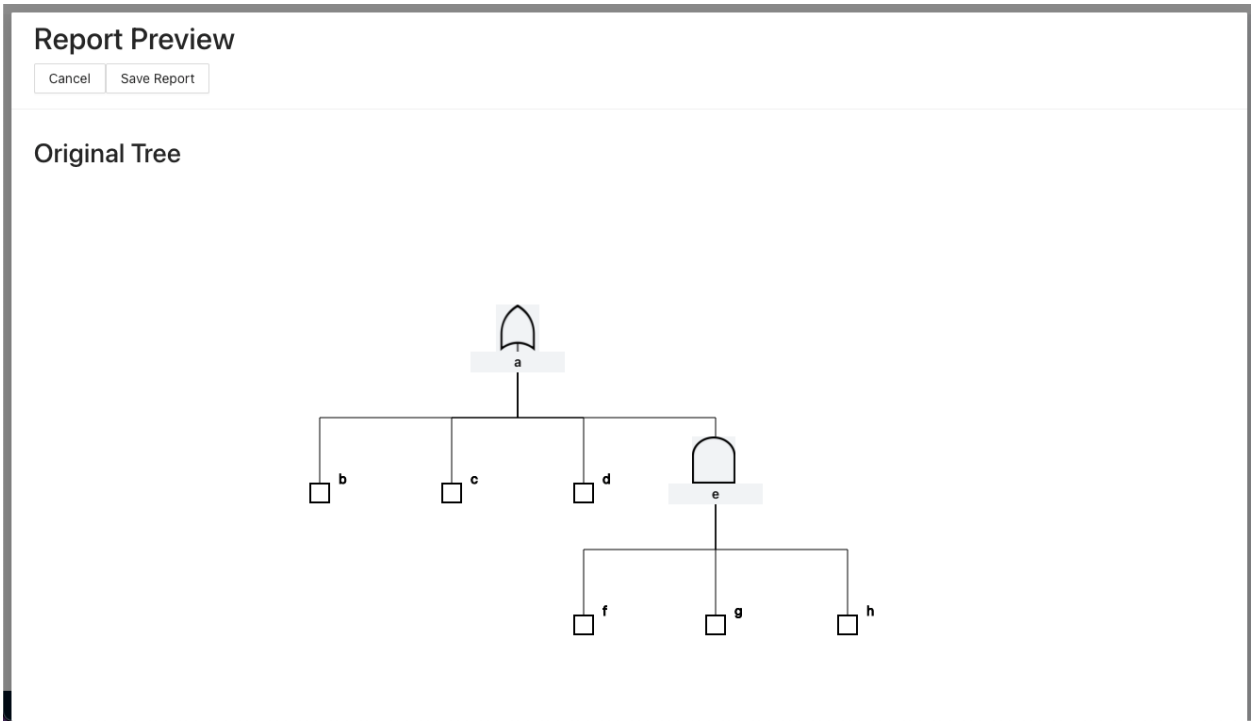


Figure 49: Partial view of Report Preview

Step 3: Select “Save Report”

Step 3: Choose destination folder

Step 4: Filename name must end with .html

Step 5: Click “Save”

Final State:

The report is exported as a .html file in your destination folder

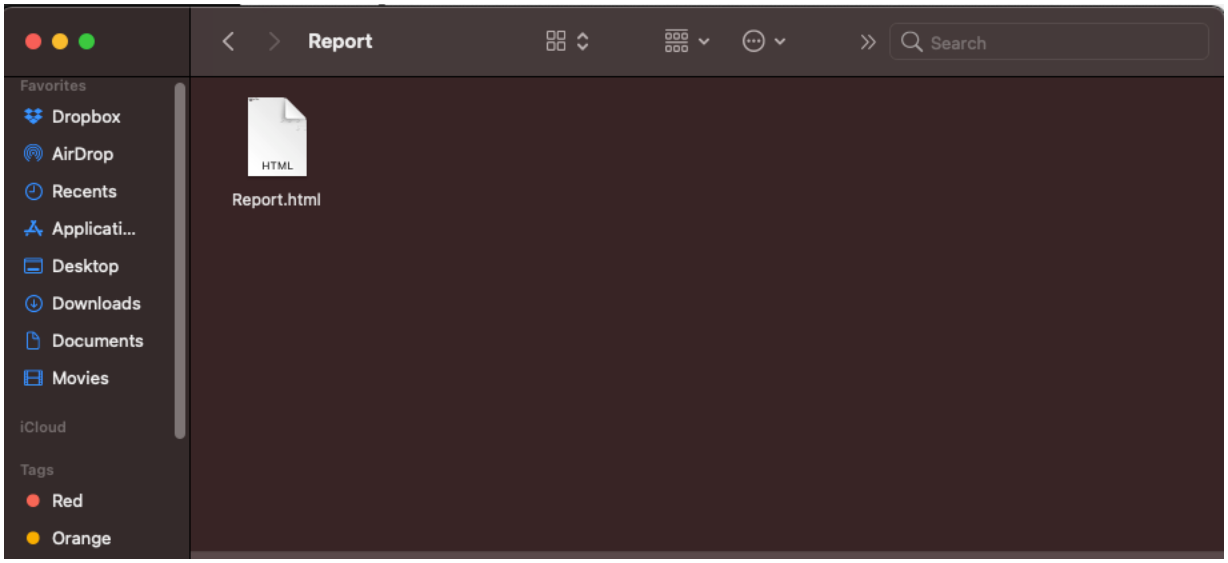
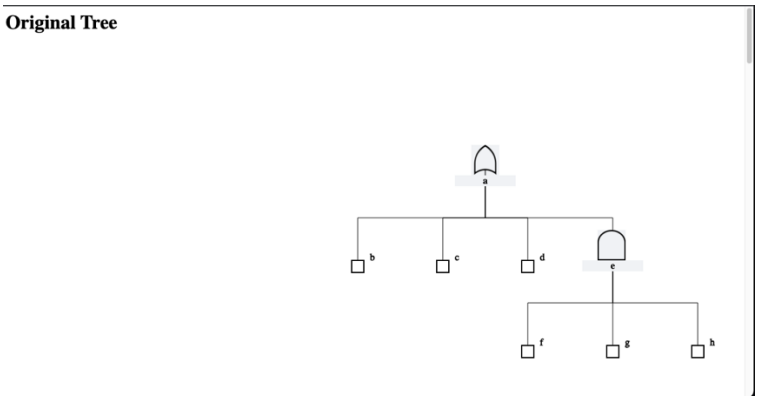
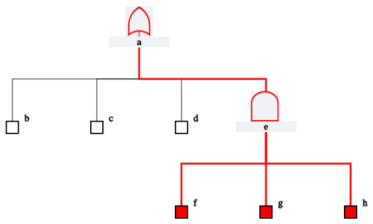


Figure 51: A .html file found in your desired directory

The following set of screenshots present the generated report (Report.html) on a browser:



Scenario 1

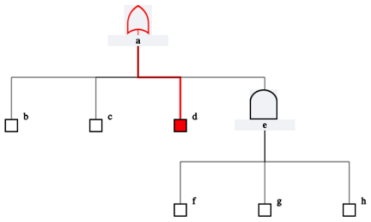


Recommendations for Scenario 1

Highest Metrics to be Mitigated:

- Node "g" with a Likelihood of 0.8
- Node "h" with a Victim Impact of 0.7
- Node "f" with a Risk of 0.8
- Node "g" with a Time Difficulty Ratio of 0.4

Scenario 2

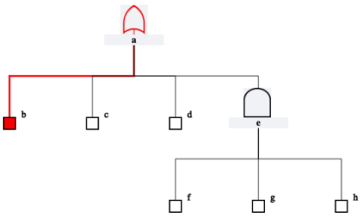


Recommendations for Scenario 2

Highest Metrics to be Mitigated:

- Node "d" with a Likelihood of 0.3
- Node "d" with a Victim Impact of 0.5
- Node "d" with a Risk of 0.4
- Node "d" with a Time Difficulty Ratio of 0.9

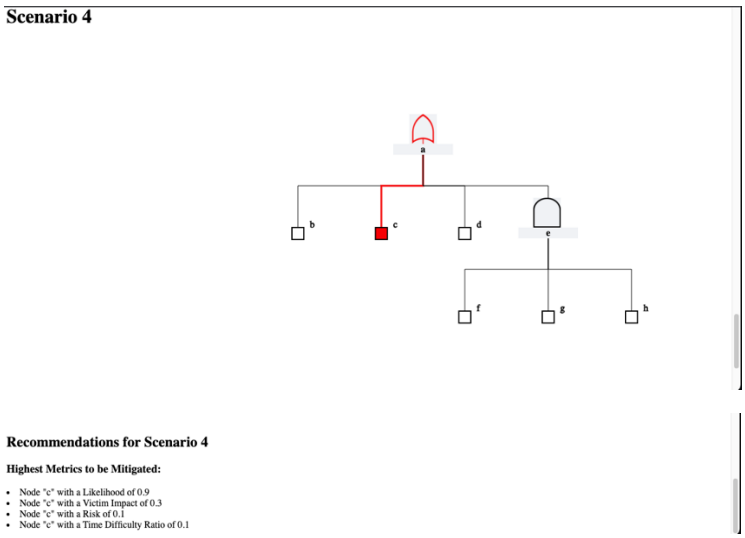
Scenario 3



Recommendations for Scenario 3

Highest Metrics to be Mitigated:

- Node "b" with a Likelihood of 0.7
- Node "b" with a Victim Impact of 0.1
- Node "b" with a Risk of 0.5
- Node "b" with a Time Difficulty Ratio of 0.2



Recommendations for Scenario 4

Highest Metrics to be Mitigated:

- Node "c" with a Likelihood of 0.9
- Node "c" with a Victim Impact of 0.3
- Node "c" with a Risk of 0.1
- Node "c" with a Time Difficulty Ratio of 0.1

Figure 52: A series of screenshots to display the generated report.