# Report for Assignment 1

## Project chosen

Name: bandit

URL: <https://github.com/PyCQA/bandit>

Number of lines of code and the tool used to count it: 20,275, lizard

Programming language: Python

## Coverage measurement: 94%, 5588 statements, 345 of which are not covered

### Existing tool

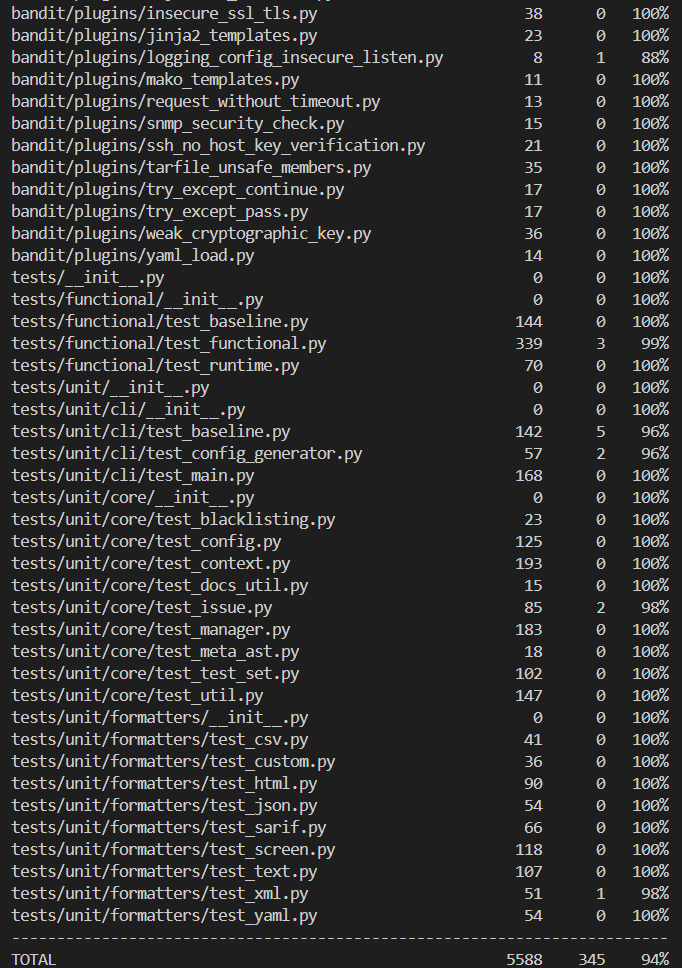
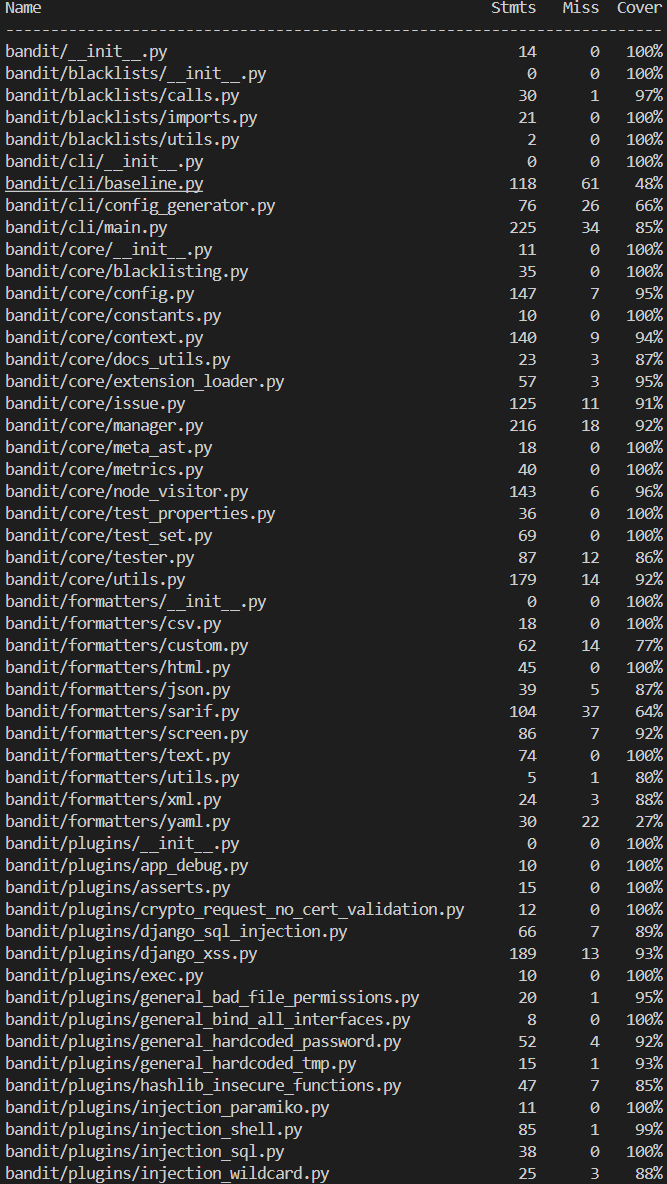
The coverage was acquired through the use of coverage.py (<https://coverage.readthedocs.io/en/7.5.3/>), in combination with pytest’s automatic unit test infrastructure.

In order to avoid coverage.py from also scanning installed libraries, we created a .coveragerc file in the root directory of the project, with the code:

| [run] omit =  /usr/\* |
| --- |

From there, the command “coverage run -m pytest” can be run in the root directory.

To view the results, one can either run the “coverage report” command to view a brief report of the coverage in the terminal, or “coverage html” to generate a folder called “htmlcov”, inside of which an “index.html” file is located. Opening this file in a browser gives a much more in-depth look at the coverage. Both of these commands are also run in the project’s root directory.

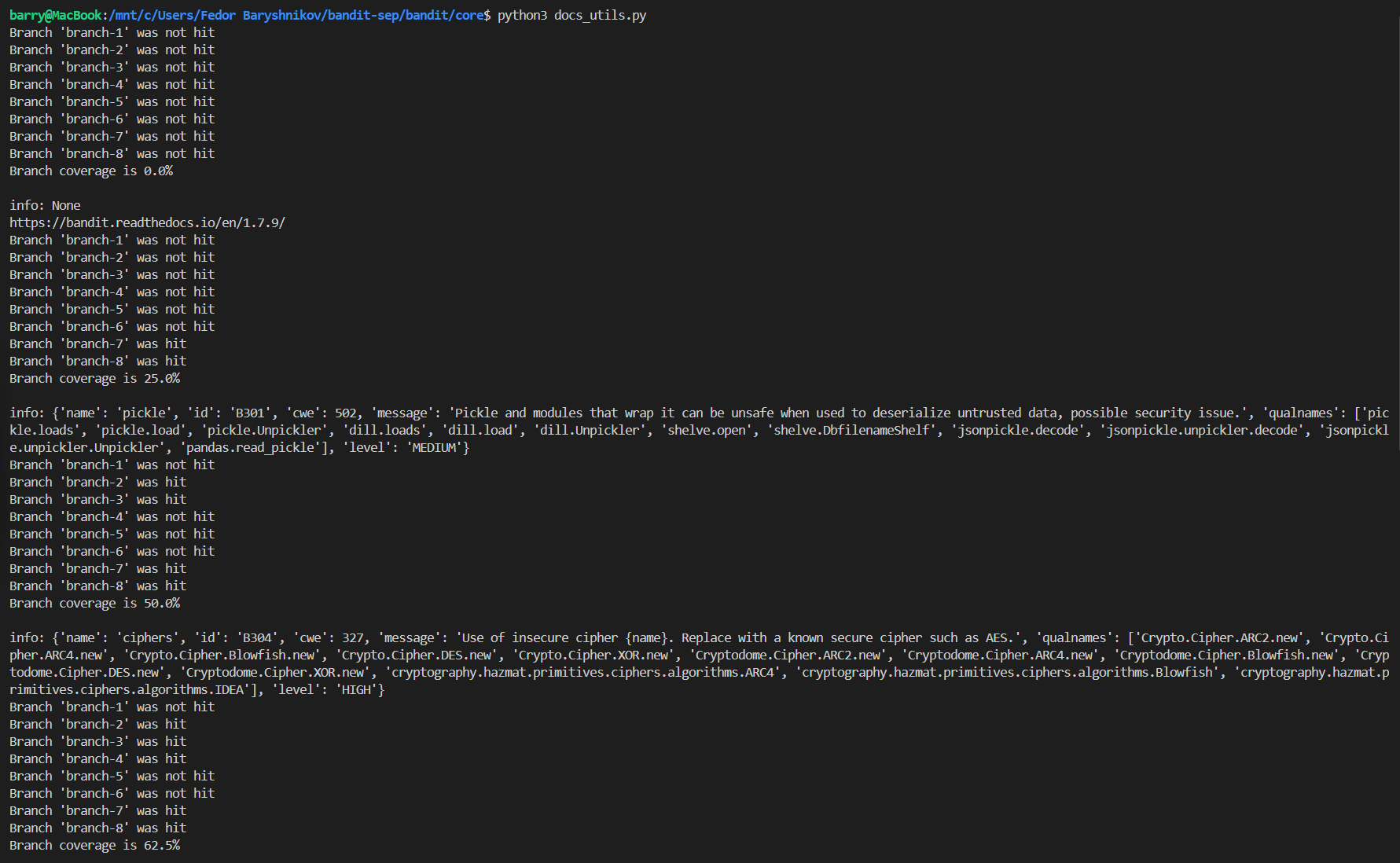


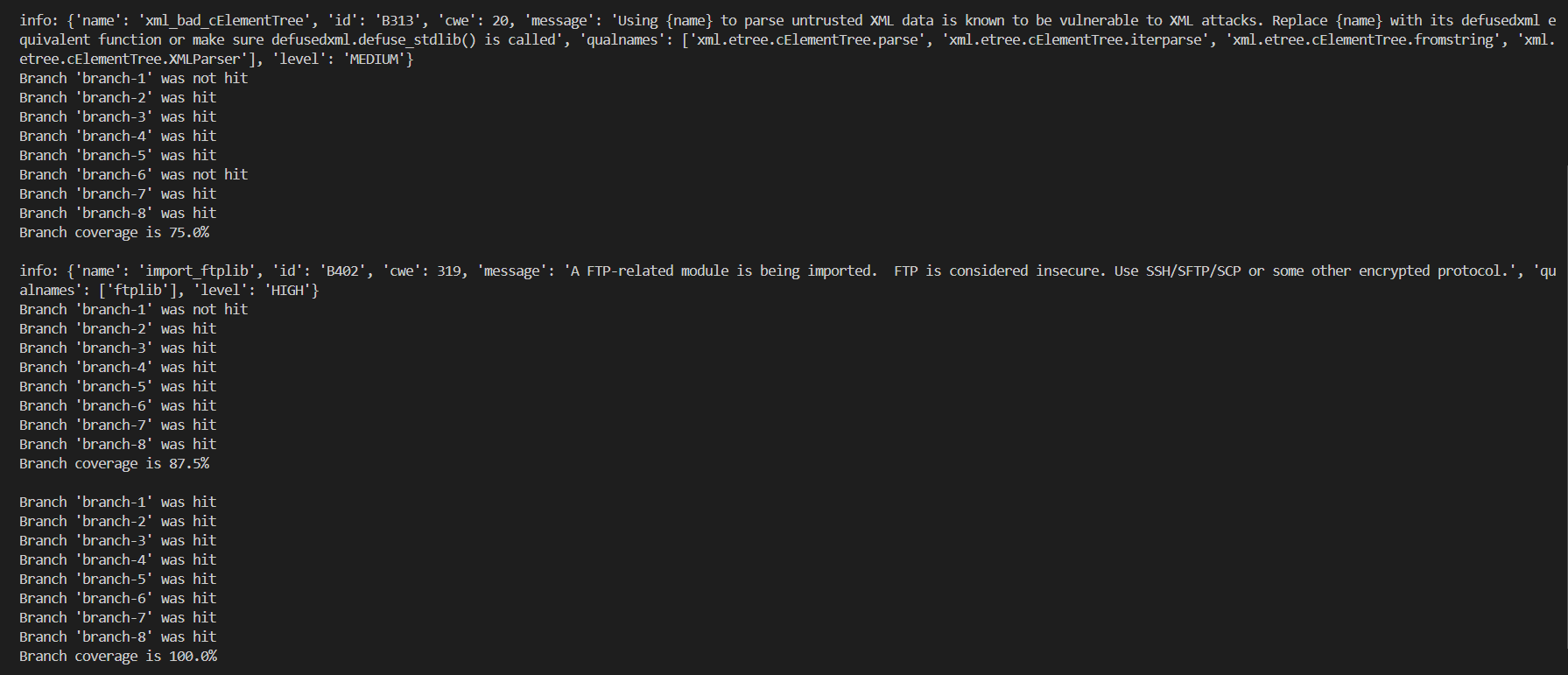
### Your own coverage tool

<Fedor>

<get\_url()>

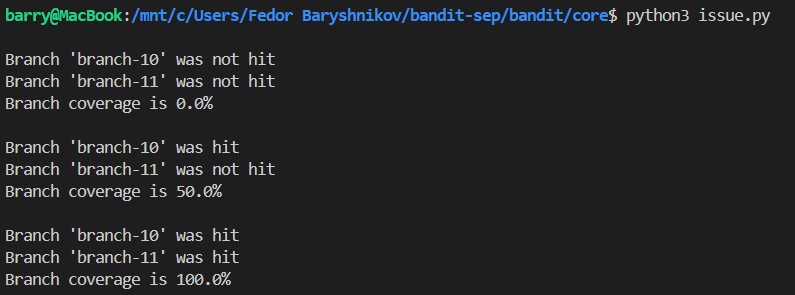
<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/8362a5ee800d797e430011d0abacaff4e6402368>>





<from\_dict()>

<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/0167bfd336b64d1d7d31ac031d99a2a7c217d22b>>



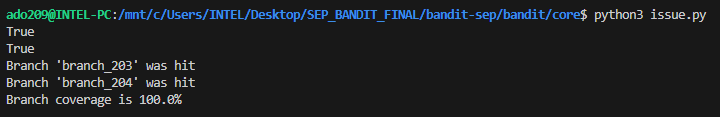
<Dogaru Eduard-Alexandru>

< Function 1: get\_code() >

< Commit made in our forked repository that shows the instrumented code to gather coverage measurements:

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/07b86a2e9fced6d9d041fbae2c67f55c1c08486b> >

< Screenshot with the coverage results output by the instrumentation: >

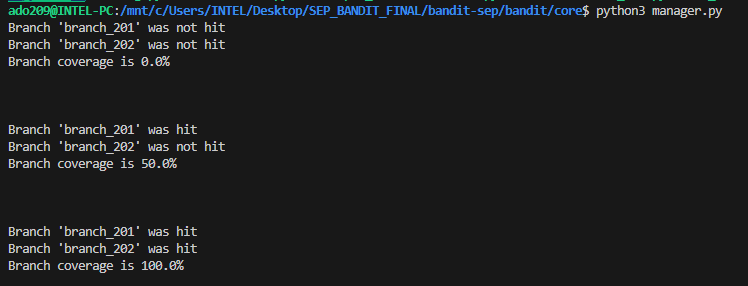


<Function 2: get\_skipped()>

< Commit made in our forked repository that shows the instrumented code to gather coverage measurements:

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/15382128b918ce84f61591cac916e1316ac81358>>

< Screenshot with the coverage results output by the instrumentation: >



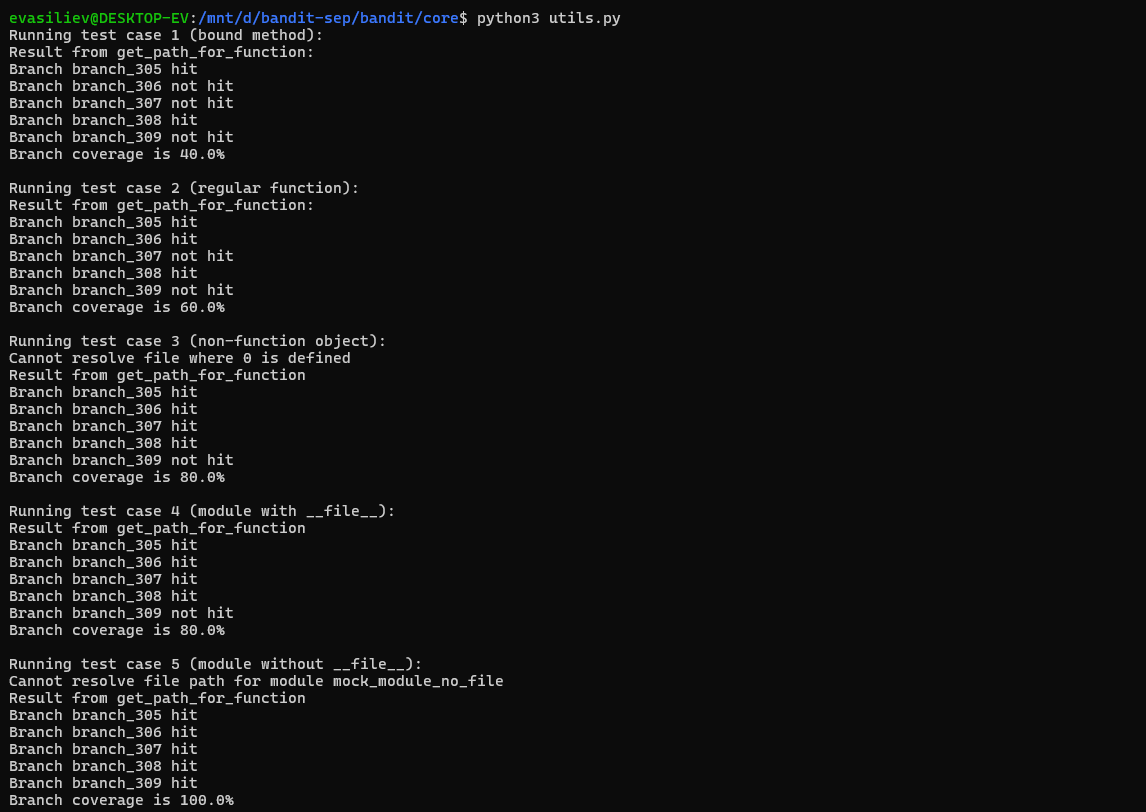
<Egor Vasiliev>

<Function 1: get\_path\_for\_function(f)>

<Link below will show the function with instrumentation, according to the task description

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/109afb5d60bdae0db1c5db4555584b9c0f4f399c#diff-8b9cca5f018d08c9437923faa3d1632bfbcc10c658187bce59b7ff658e5fd27b>>

< Screenshot with the coverage results output by the instrumentation: >

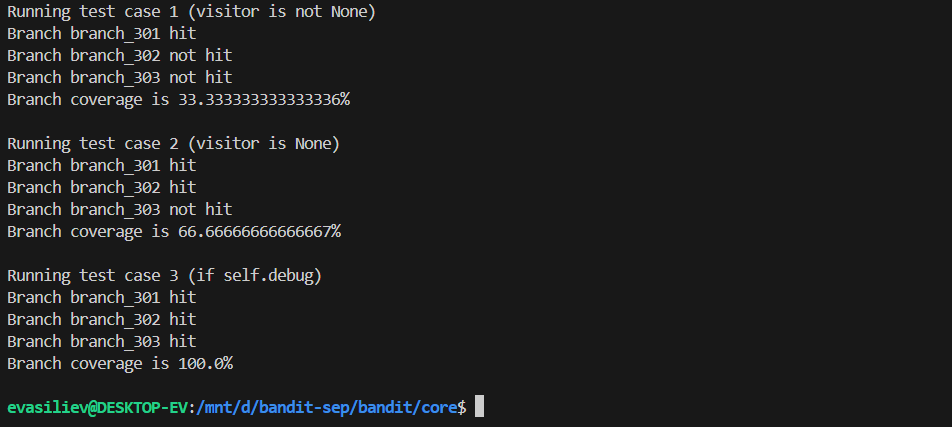


<Function 2: visit(self, node)>

<Link below will show the function with instrumentation, according to the task description

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/109afb5d60bdae0db1c5db4555584b9c0f4f399c#diff-7a4dd8060f48ec6915b6ae228c29c575ba25b5f067a506d6e1398a902845ceff> >

< Screenshot with the coverage results output by the instrumentation: >

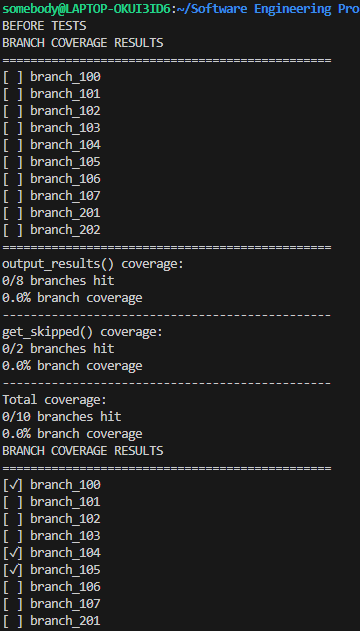


<David (Jia Hao) Ji>

<output\_results()>

<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/687b8e59f374b5d5c2bd0c523108db8af98fe8a7#diff-45c7c0a49c4b96875ac4395e3c1a82601ad9454e278ec0537e6b3f1b8235c1ff>>

<Provide a screenshot of the coverage results output by the instrumentation>



<get\_test\_id()>

<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/687b8e59f374b5d5c2bd0c523108db8af98fe8a7#diff-4e949bddd9e1af64c746ad995c0b2b4dae1a008f194b84b7c42244eeff5b3664>>

## Coverage improvement

### Individual tests

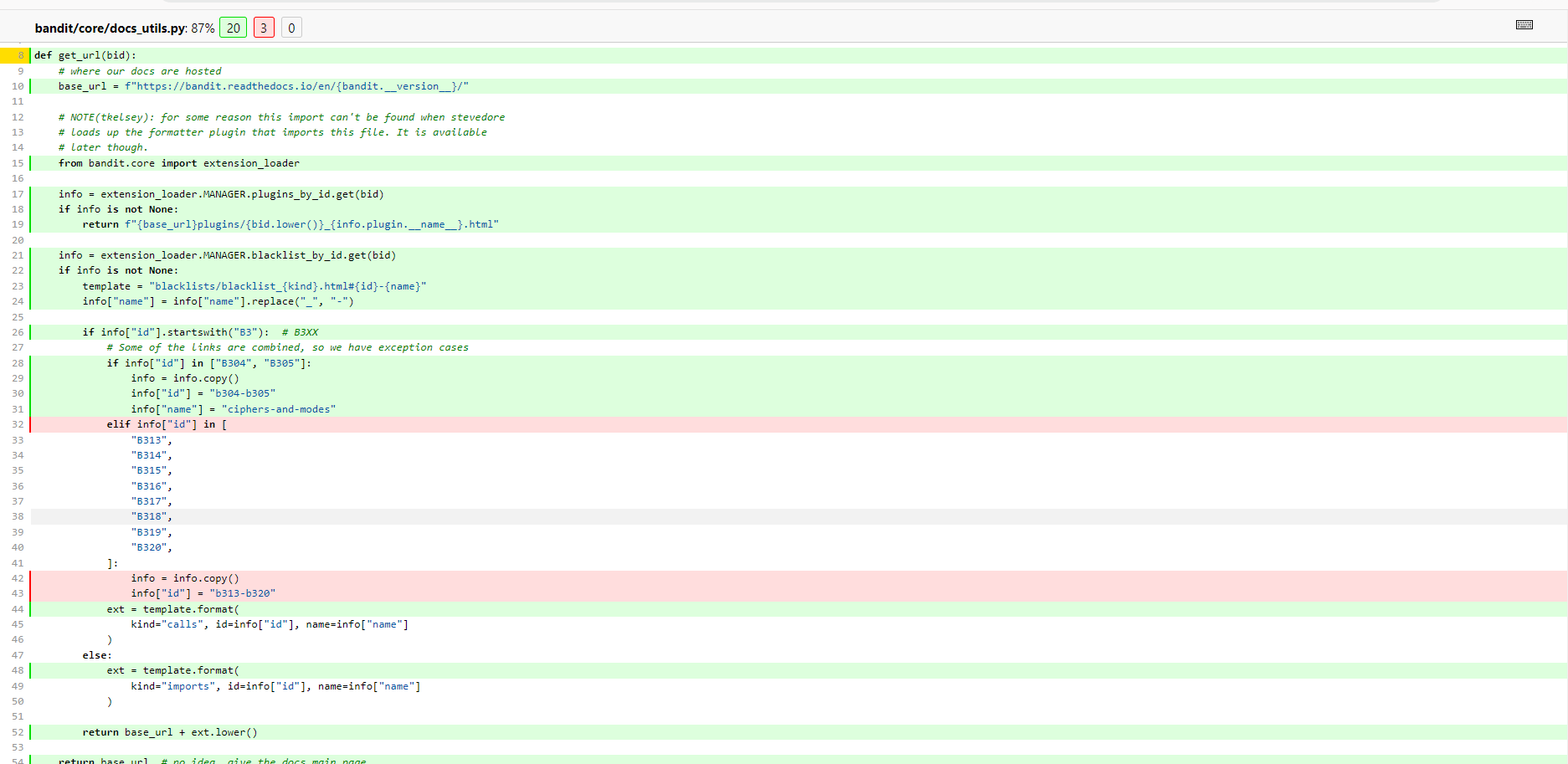
<**Fedor Baryshnikov**>

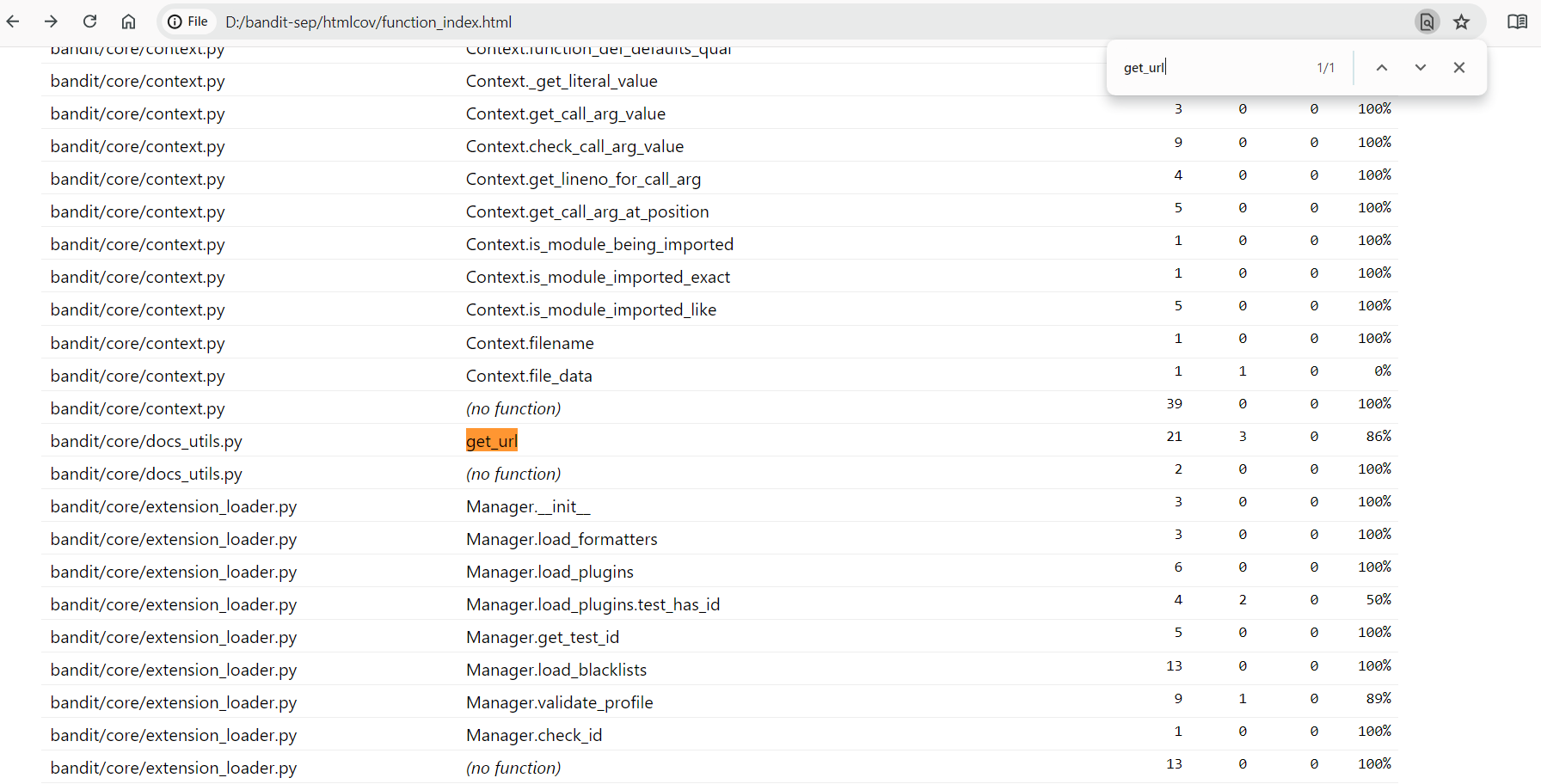
<test\_invalid\_id>

<Link to a commit made in the forked repository that shows the new test:

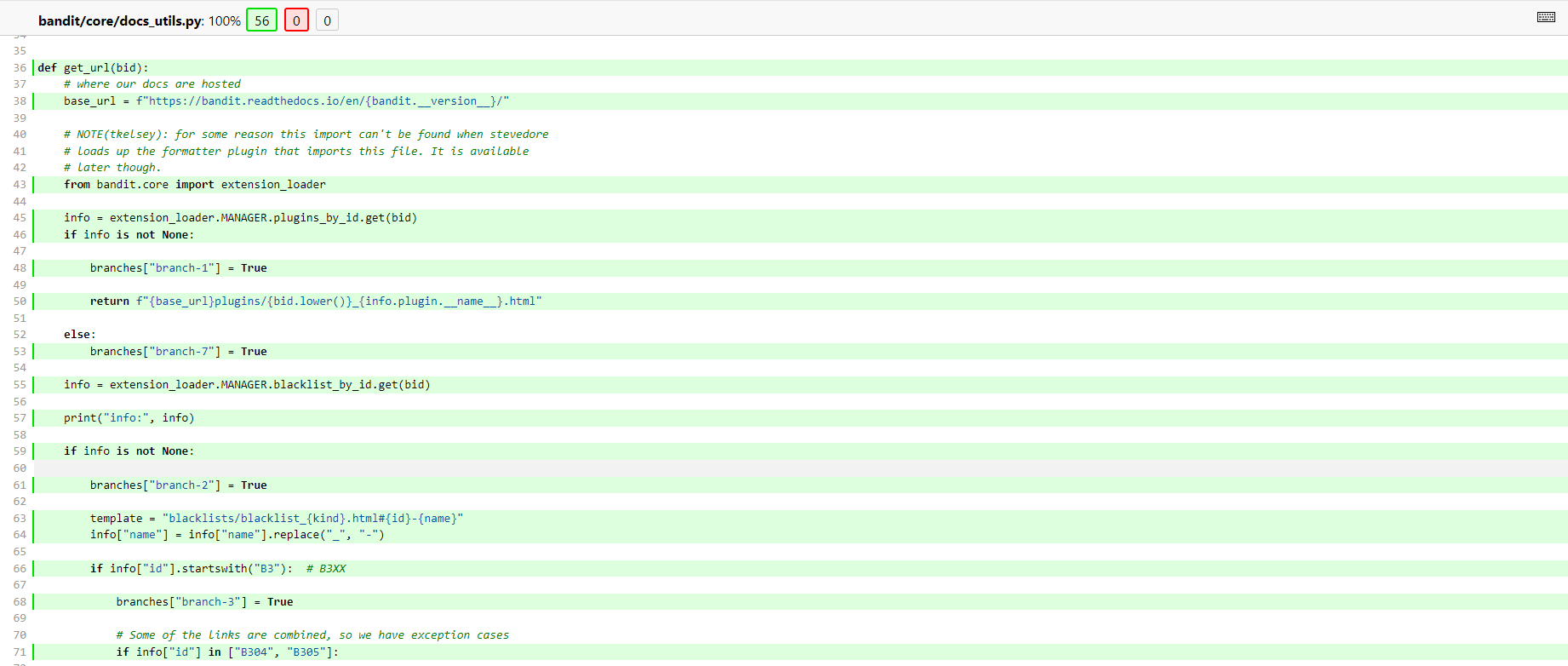
<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/c02cd09be149985850e5ed91656eeb7b22b9e07c#diff-347c41a83d86f15e8182a5b55829cae18c93586610a6798a74e18f4f59467278> t>

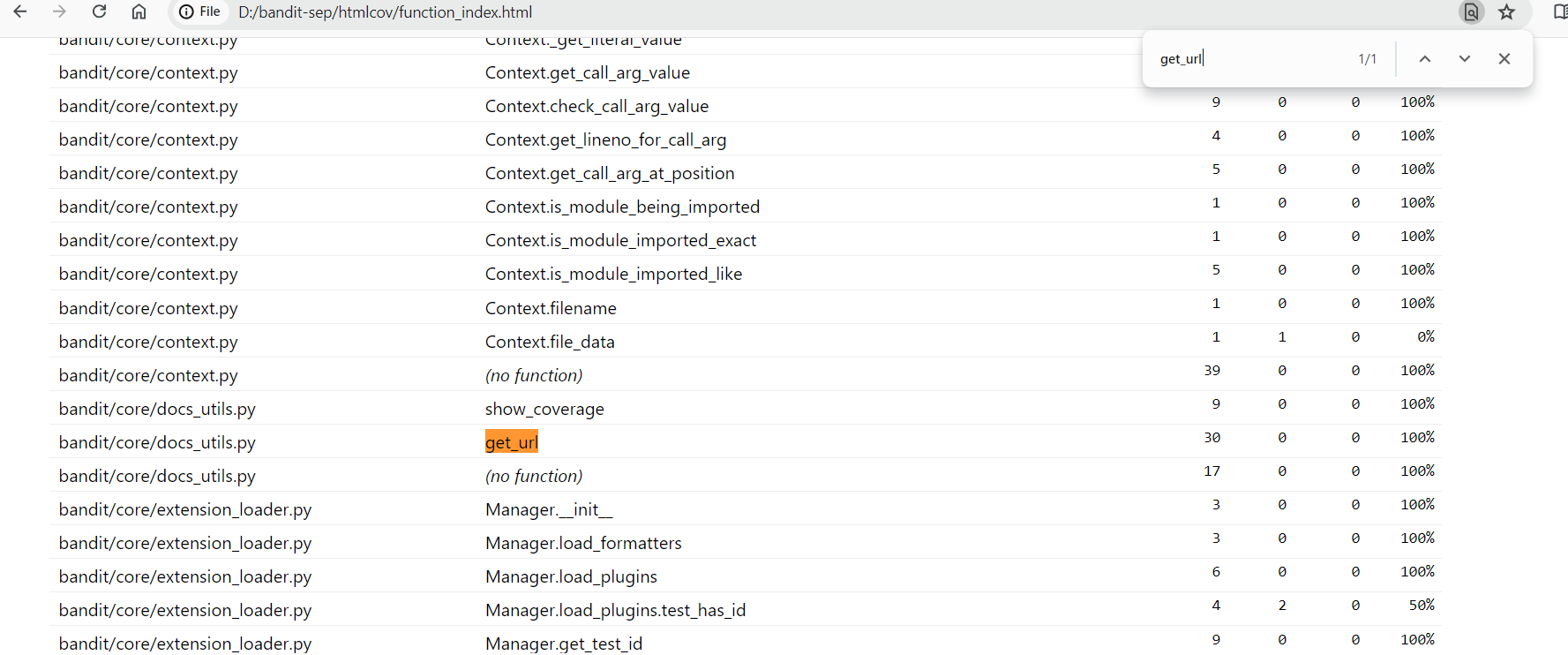
<A screenshot of the old coverage results>





<A screenshot of the new coverage results>

****



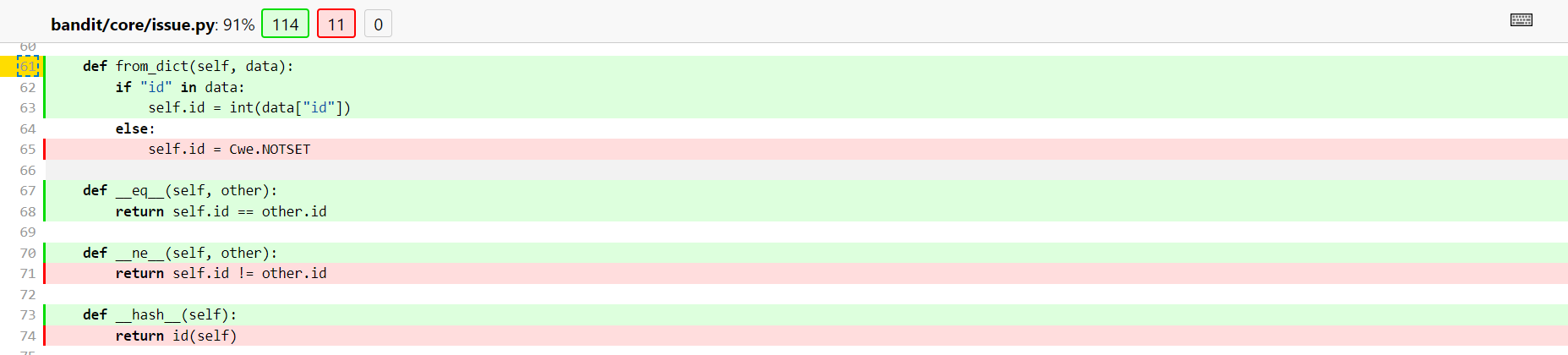
<Coverage of the original function (before any enhancements) was 86%. Four tests have been implemented in order to cover all the branches of the get\_url() function. These are: test\_invalid\_id, test\_startswith\_B3, test\_in\_range\_B313\_B320, and test\_startswithB\_first\_digit\_not\_3. After all these implemented tests, the coverage of the get\_url() function reached a total of 100% which in turn meets the requirements of the task.

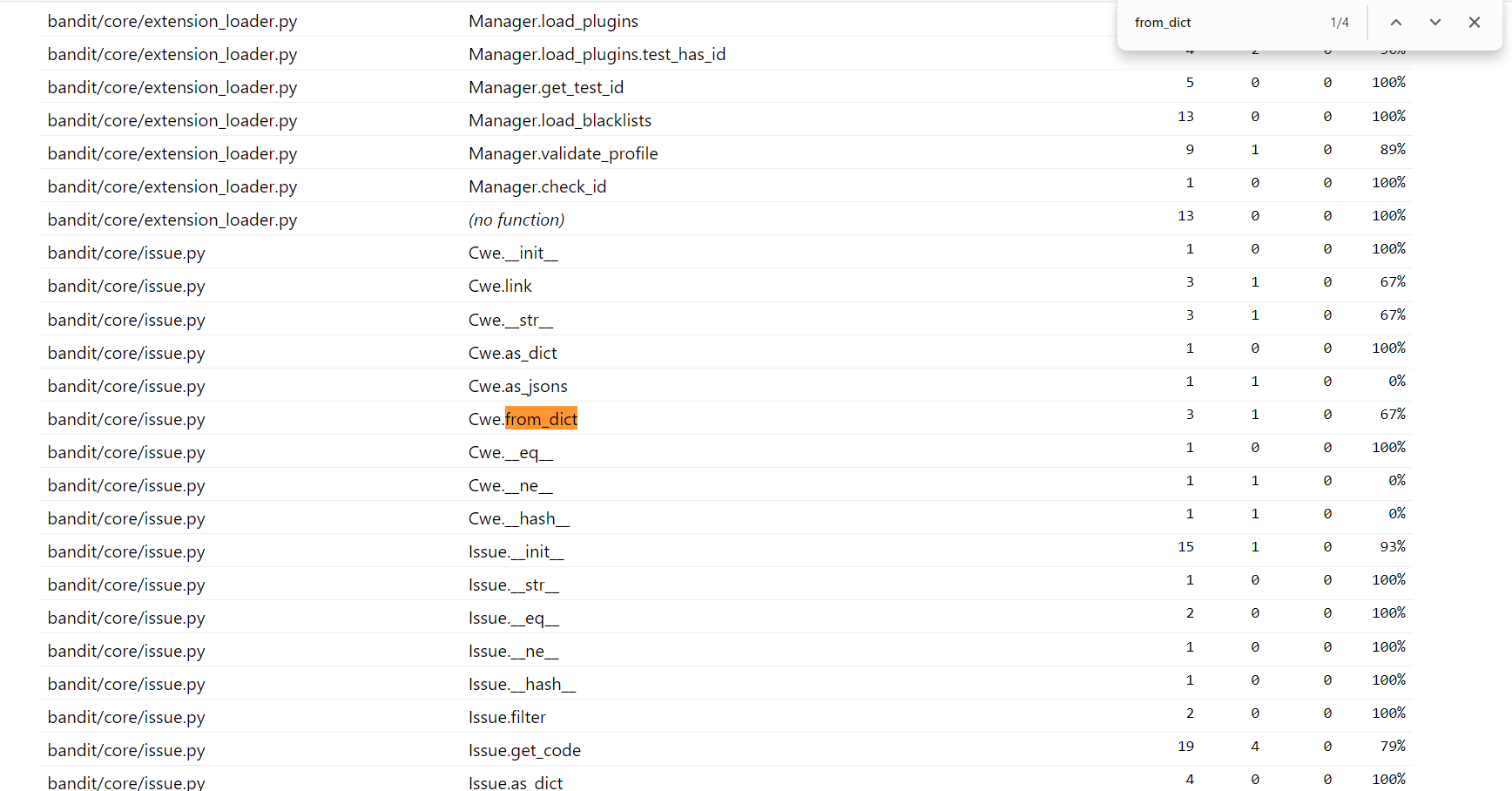
<test\_valid\_id>

<Link to a commit made in the forked repository that shows the new tests:

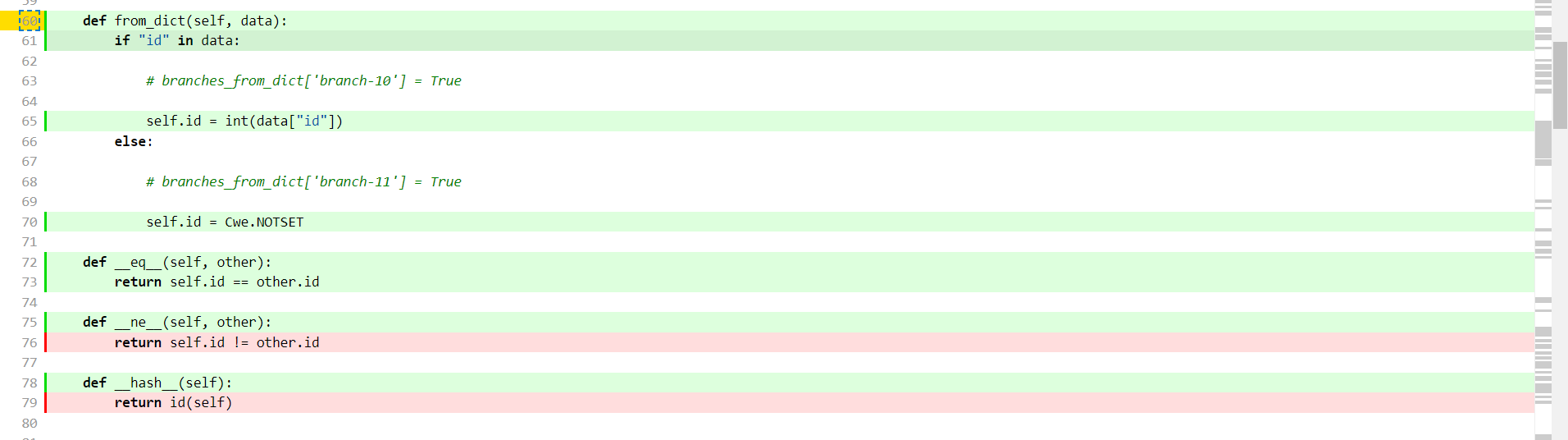
<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/c02cd09be149985850e5ed91656eeb7b22b9e07c#diff-f6516330e75983d1cd33746823f5783eb5e7e160756616654ce467620260b2c6>>

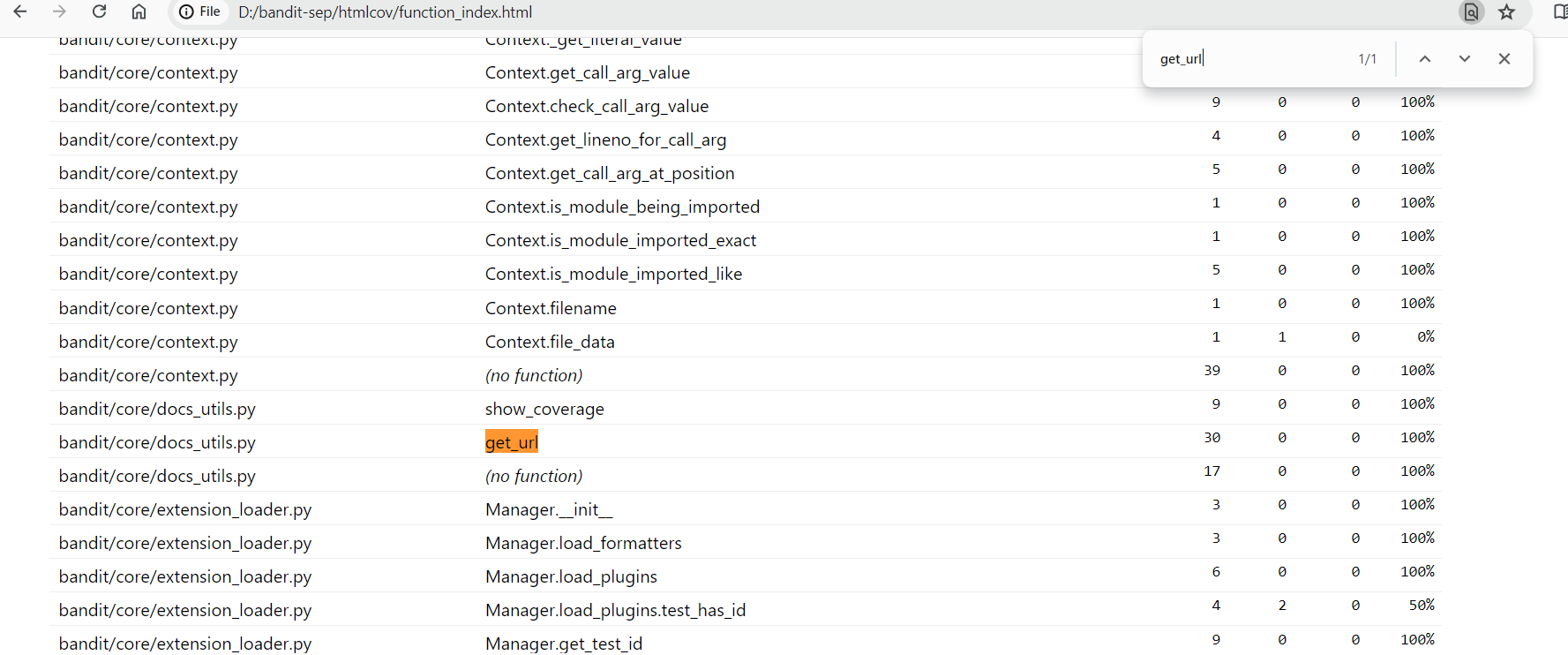
<A screenshot of the old coverage results>





<A screenshot of the new coverage results>





<Initial unit tests covered from\_dict() function only by 67%. After implementing a new test case the coverage of the mentioned function increased by 33% reaching a total of 100%. In order to get improved coverage and to cover all the statements in the investigated function 2 tests were implemented, specifically test\_valid\_id and test\_invalid\_id. These tests ensure both scenarios of valid and invalid id handing from from\_dict() function.>

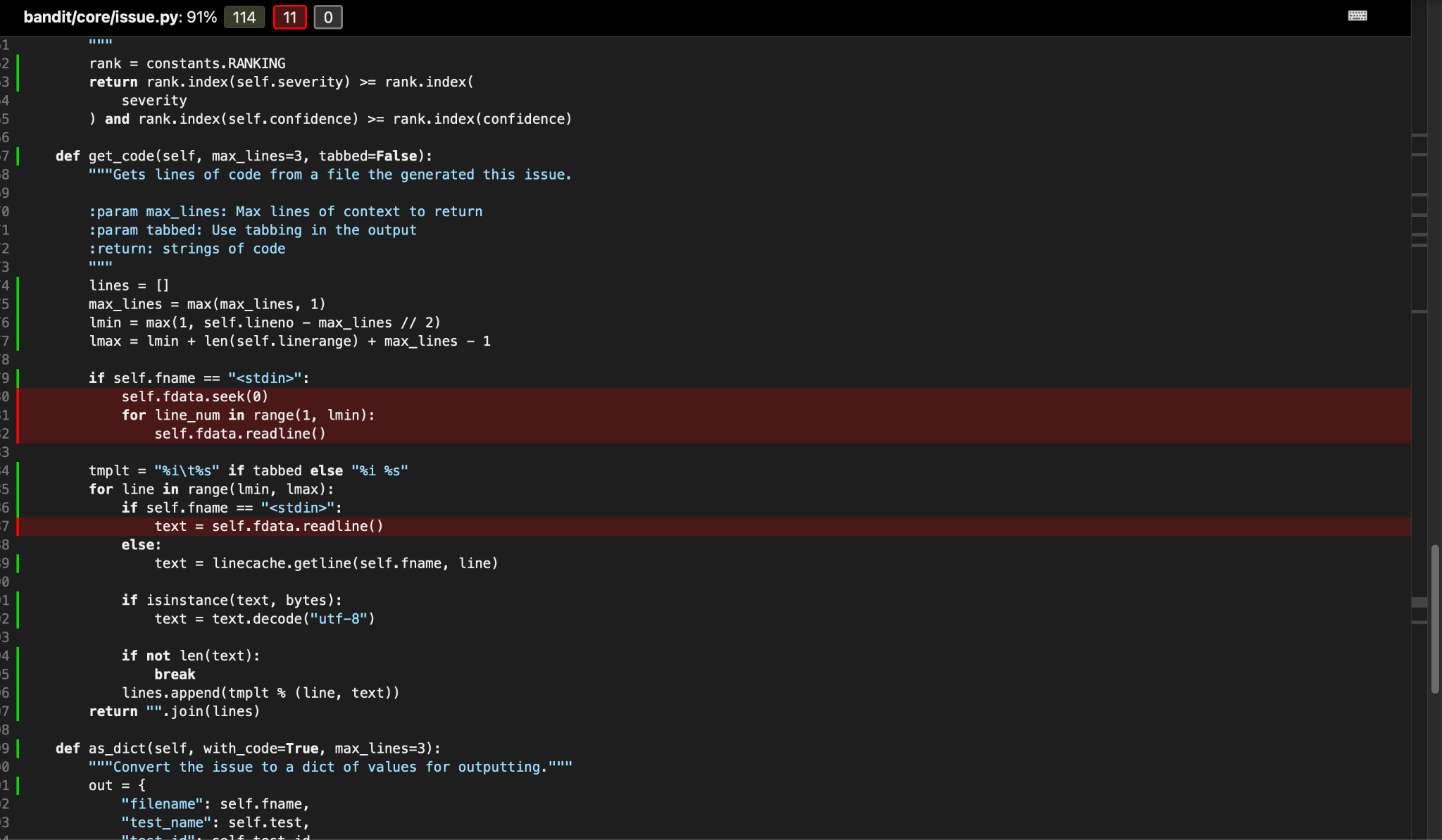
<Dogaru Eduard-Alexandru>

<Test 1: test\_get\_code>

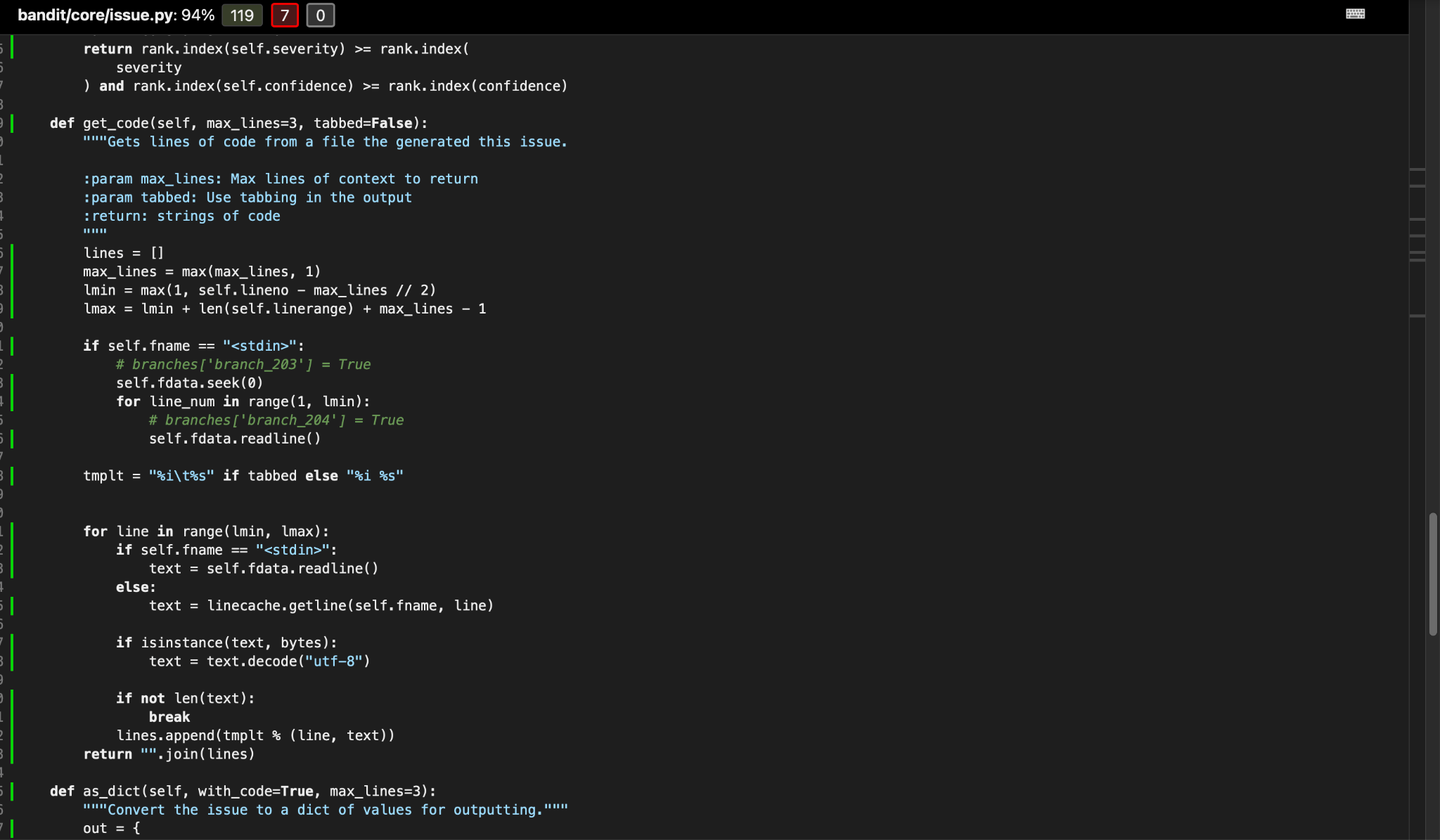
<Link to a commit made in the forked repository that shows the new test:

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/f0d52b1f1ea4277cb84b59aa1a782b602c158d50>>

<Provide a screenshot of the old coverage results (the same as you already showed above)>



<Provide a screenshot of the new coverage results>



<State the coverage improvement with a number and elaborate on why the coverage is improved>

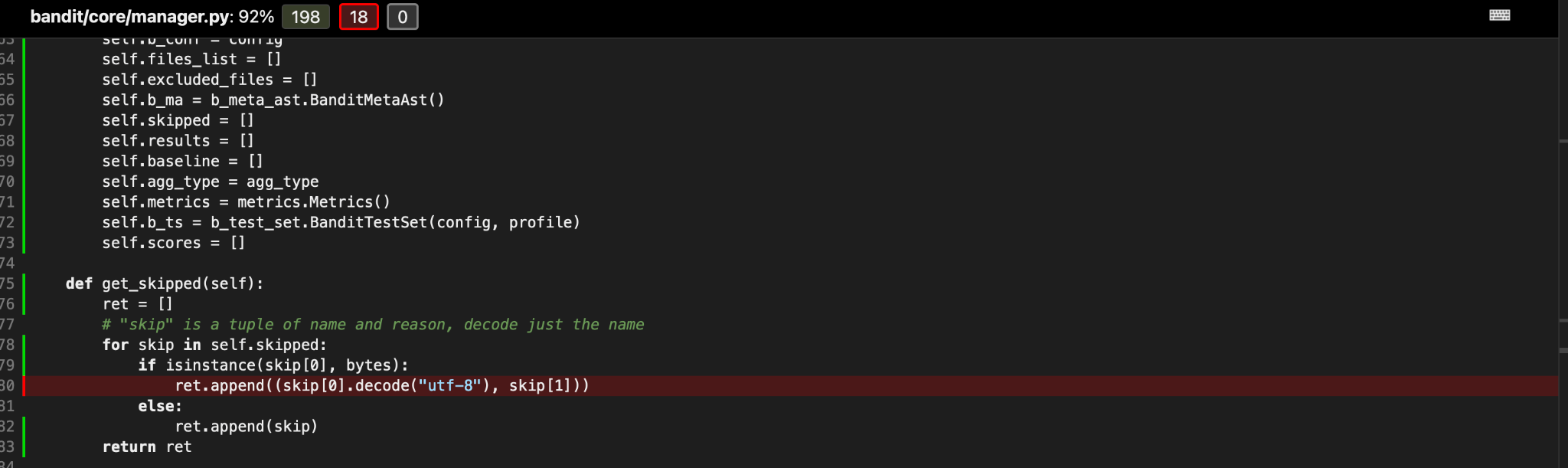
After implementing the new test, the coverage increased from 91% to 94%. Also, as depicted in the screenshots above, from 11 uncovered lines, it went down to only 7. This is because the test created covers both if statements and the for loop.

<Test 2: test\_get\_skipped\_1>

<Link to a commit made in the forked repository that shows the new test:

<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/51d514af30f7cd6d703a0c4df85d8a791313a811>>

<Provide a screenshot of the old coverage results (the same as you already showed above)>



<Provide a screenshot of the new coverage results>



<State the coverage improvement with a number and elaborate on why the coverage is improved>

The overall percentage didn’t change, even though we improved branch coverage. This is because there are too many lines which are not covered. In the second screenshot it shows that there are only 17 left (in comparison to the first screenshot, where there are 18). The coverage was improved because now both branches are covered.

<Egor Vasiliev>

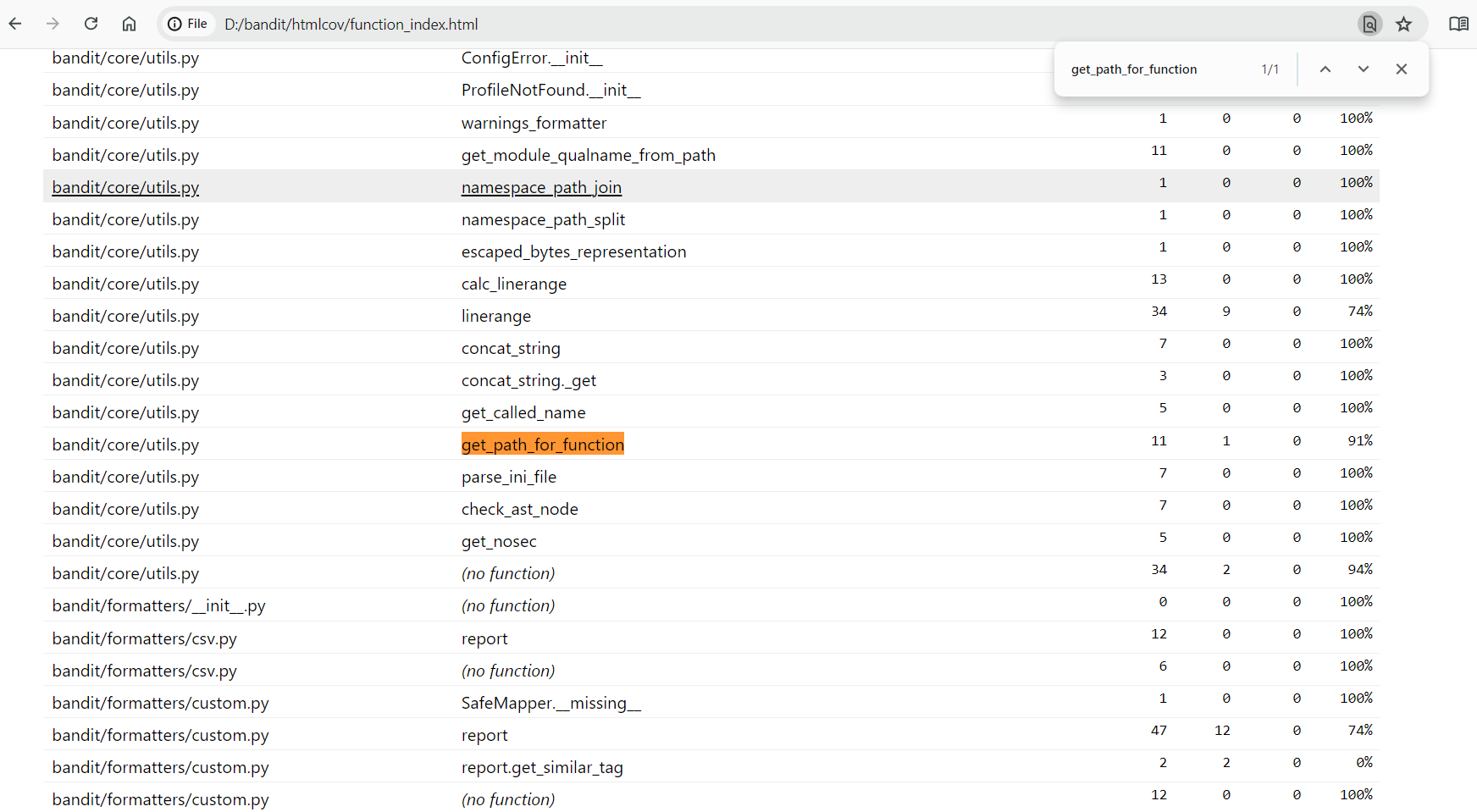
<test\_path\_for\_function\_bound\_method>

<Link to a commit made in the forked repository that shows the new test:

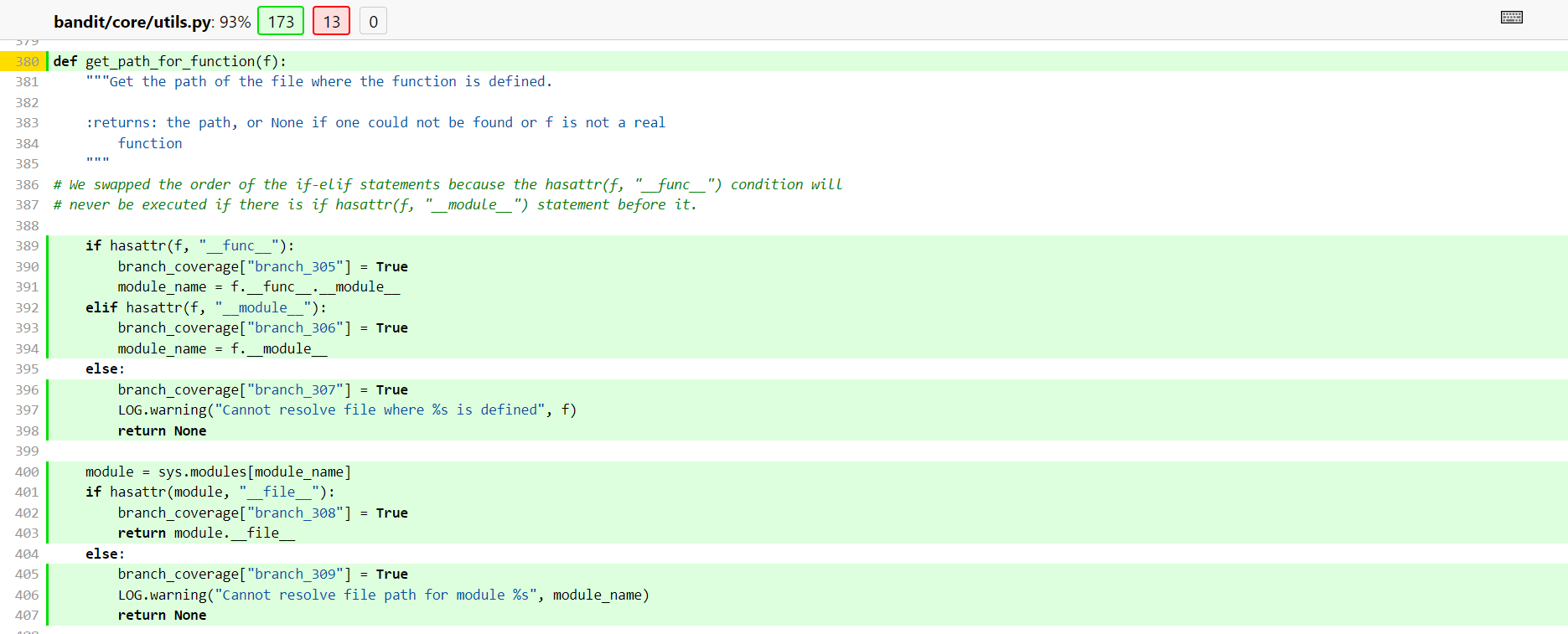
<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/dcdfcd6d09d155663af96c29ab45f608304aabf0> >

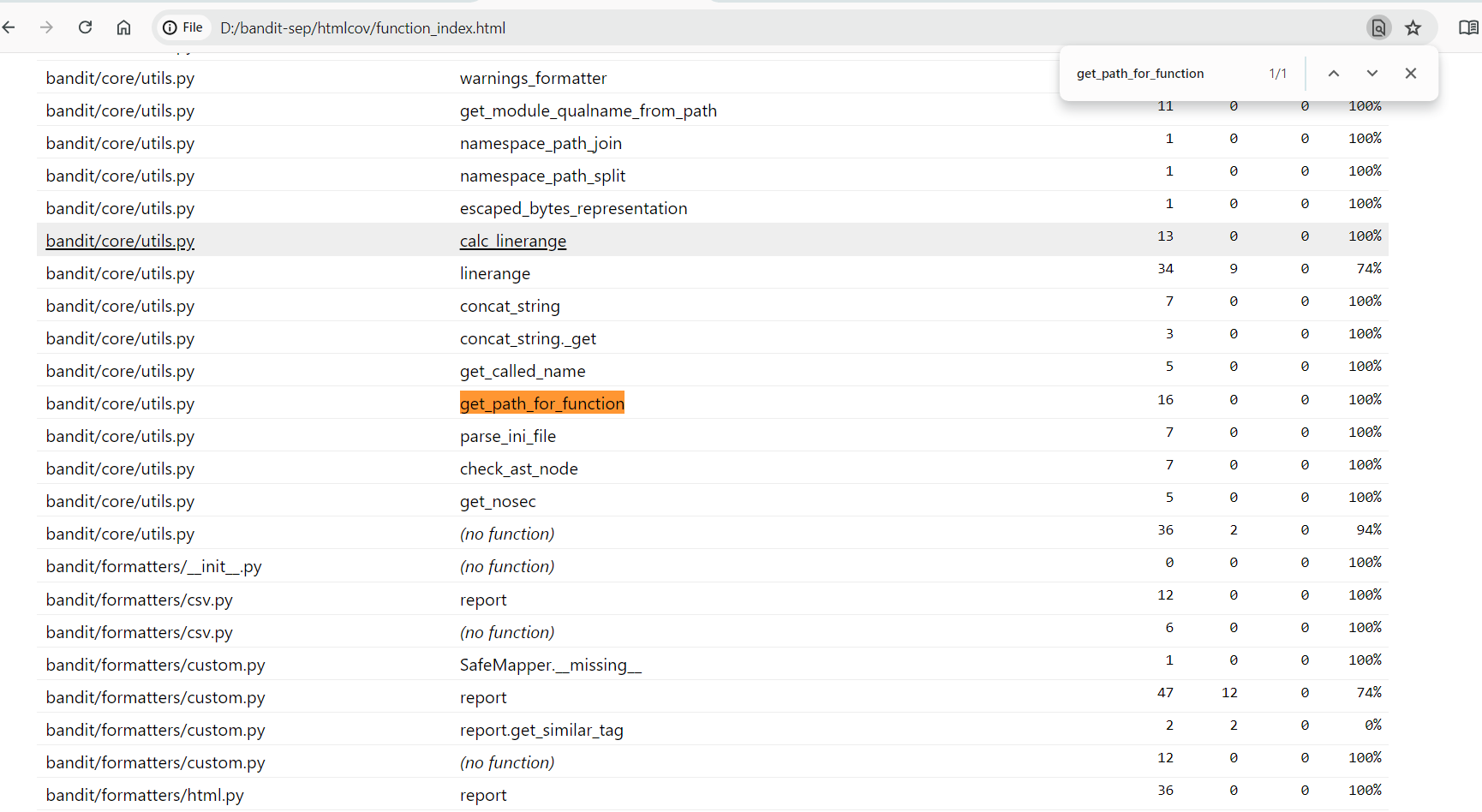
<A screenshot of the old coverage results>





<A screenshot of the new coverage results>





<As demonstrated at the screenshots coverage for get\_path\_for\_function increased from 91% to 100%. Coverage improvement was obtained by developing an additional test case, which covered all the branches in the existing get\_path\_for\_function function.

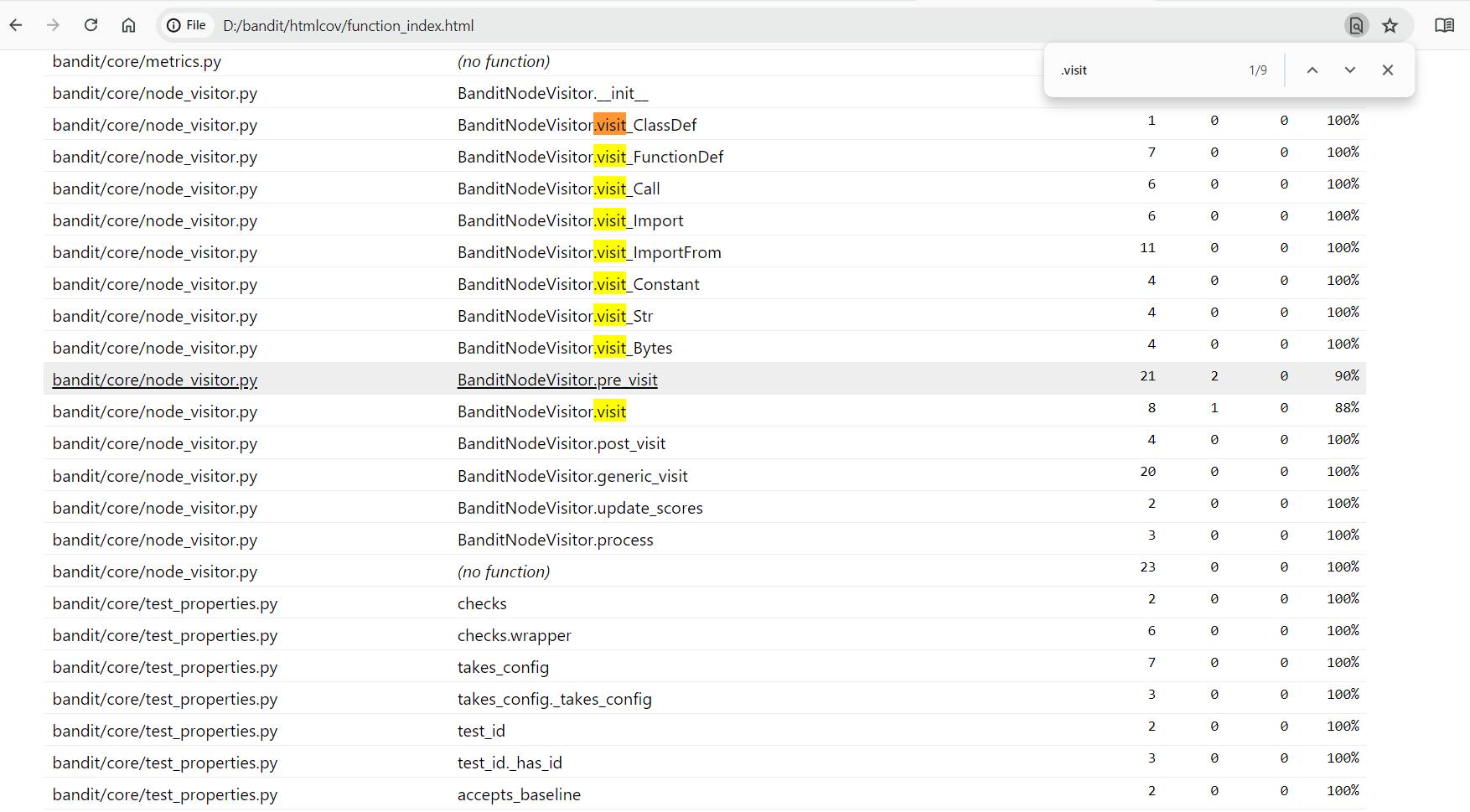
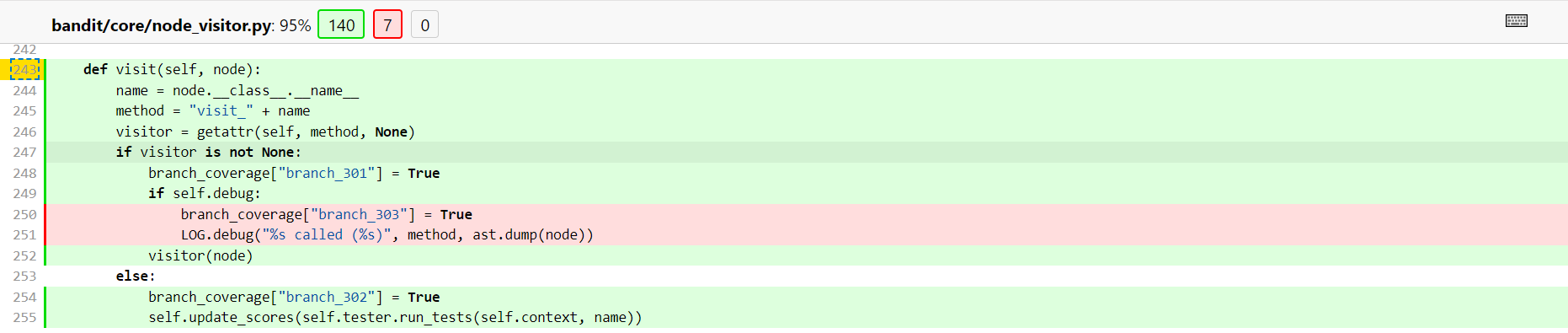
Additionally, the im\_func attribute was changed to \_\_func\_\_ (approved by professor). This was done because of the different python versions, and was necessary to avoid failures.>

<test\_visitor>

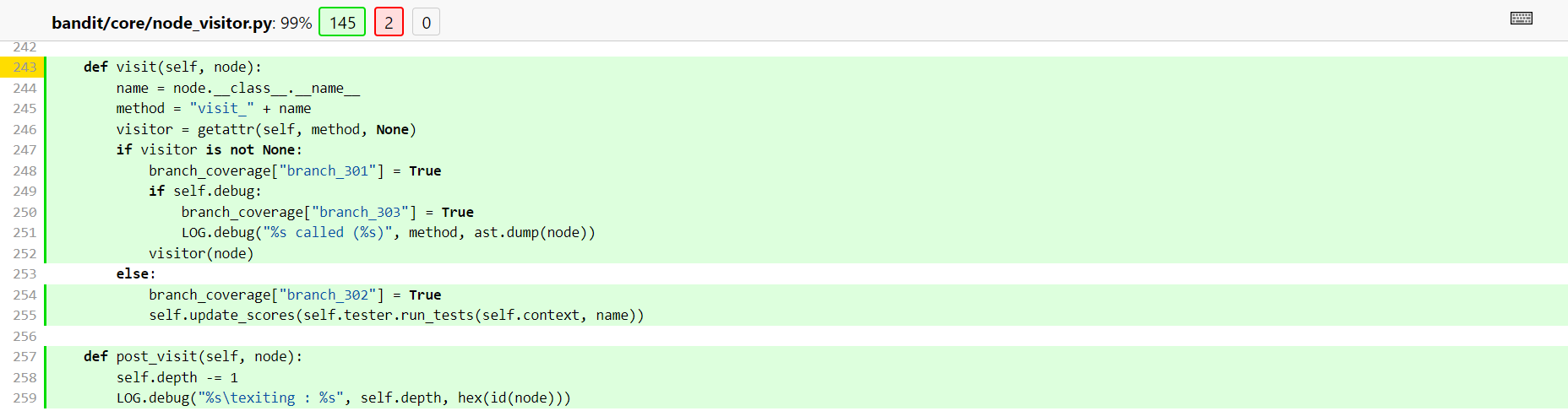
<Link to a commit made in the forked repository that shows the new test:

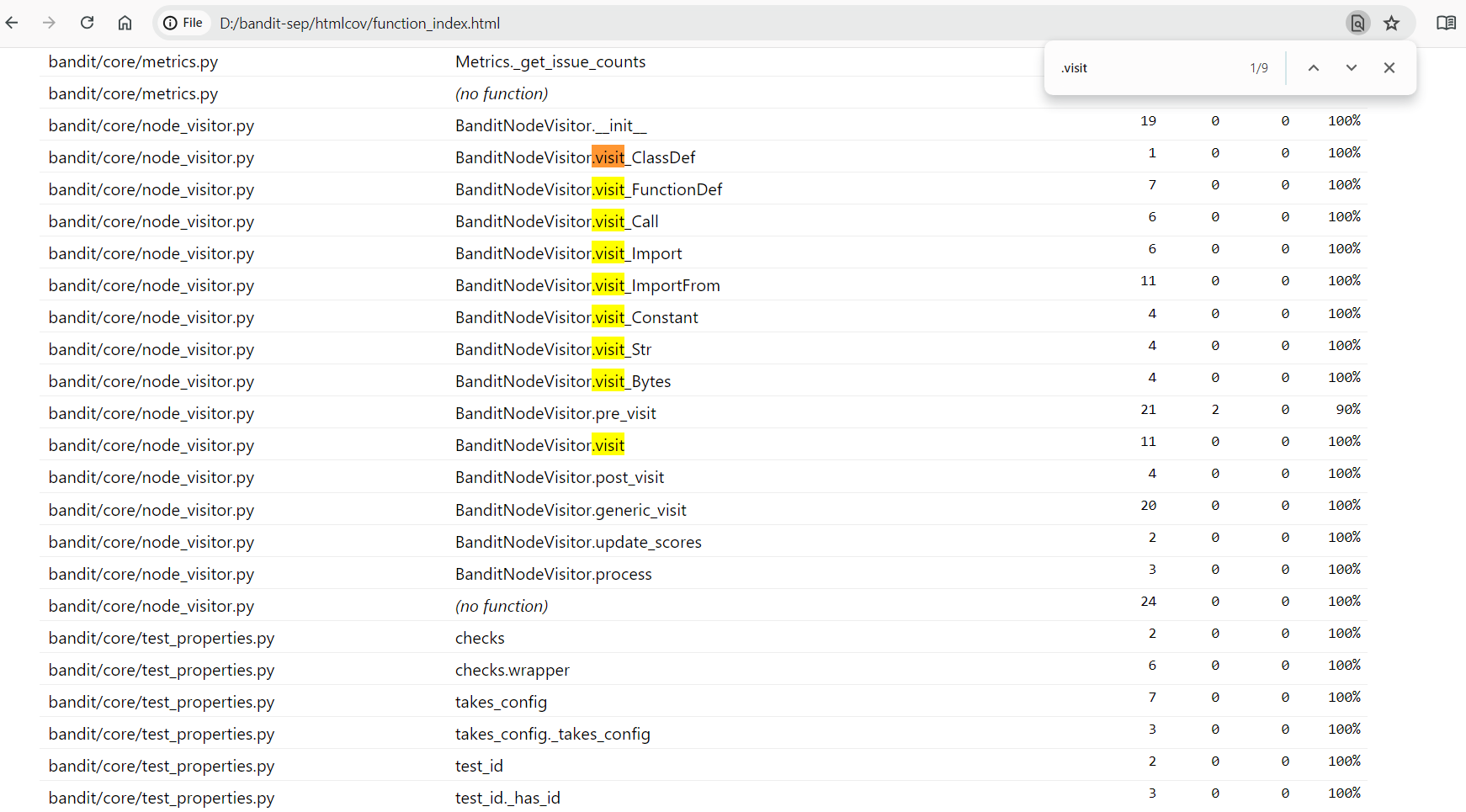
<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/cc27e3c7eb514b6d7f3174ef297710388ddb3c72> >

<A screenshot of the old coverage results>



<A screenshot of the new coverage results>





As demonstrated in the screenshots coverage for visit() function increased from 88% to 100%. Coverage improvement was obtained by developing additional test cases, which covered all the branches in the existing visit function. New implemented tests ensure that every possible scenario within the visit() function was thoroughly validated.

To get 100% coverage it was necessary to implement several additional tests, particularly tests which covered cases when: test\_visitor\_not\_none, test\_visitor\_none, test\_debug\_enabled

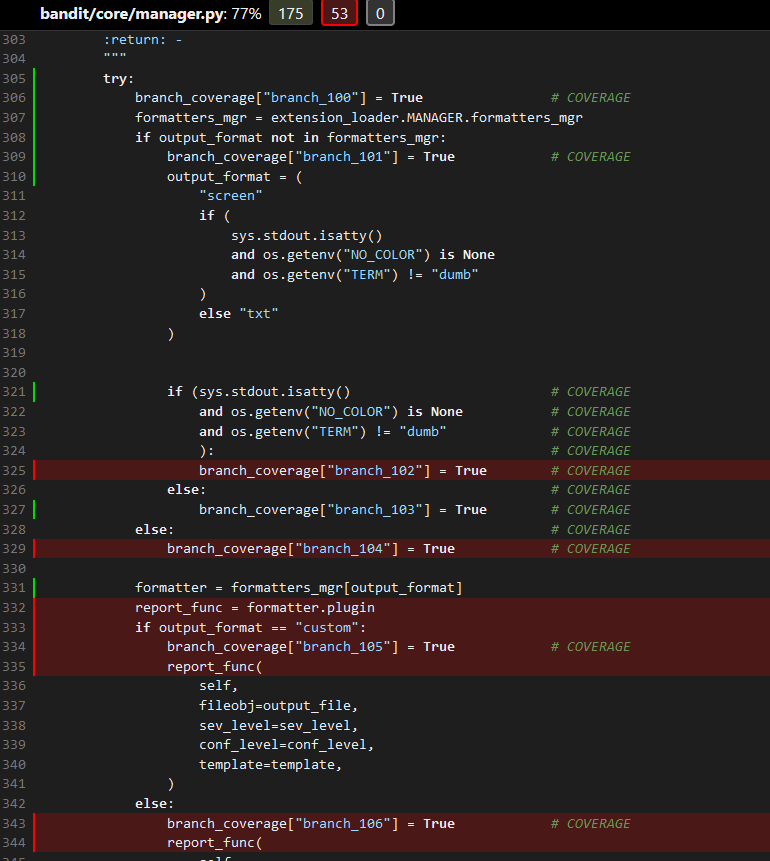
<David (Jia Hao) Ji>

<test\_manager.>

<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/687b8e59f374b5d5c2bd0c523108db8af98fe8a7#diff-70e14507056b38e9acbbc5ed966126567d46864c90e988eabe3624db8ca569c0>>

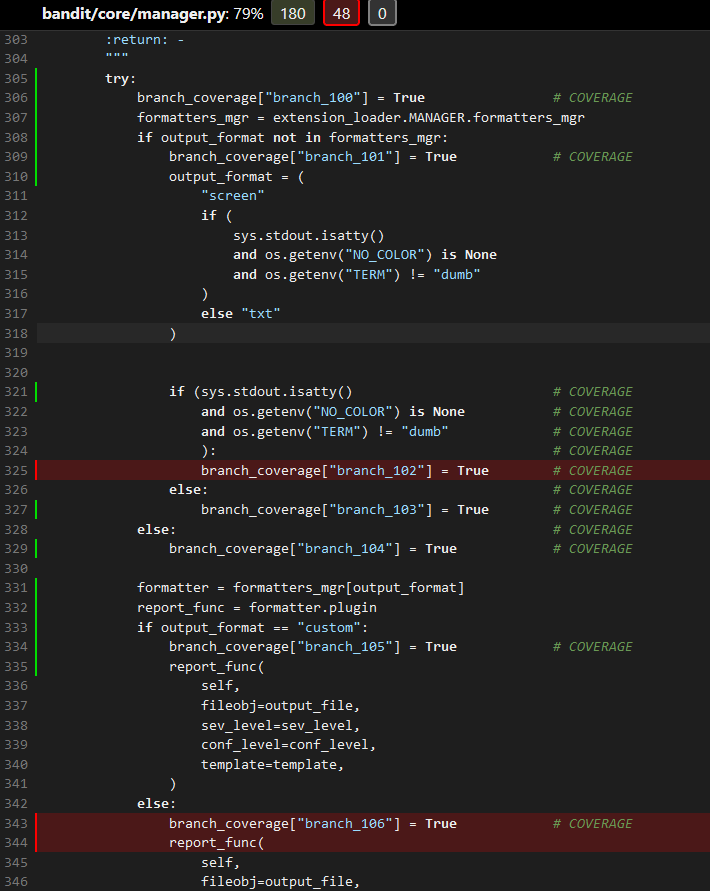
<A screenshot of the old coverage results>

**OLD COV FOR OUTPUT\_RESULT**

****

****

<Provide a screenshot of the new coverage results>

**NEW COV FOR OUTPUT\_RESULT**

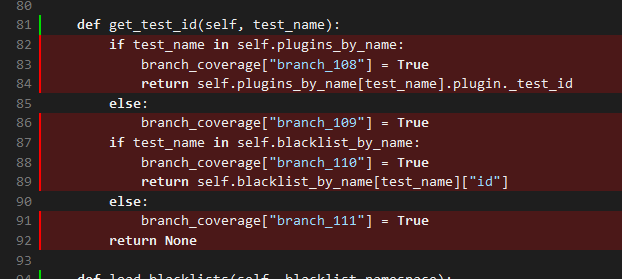


<The original function had an 77% test coverage. To achieve full coverage of all branches in the output\_result() function, new specific tests were created. These tests collectively brought the function's coverage to 79%. >

<Test 2>

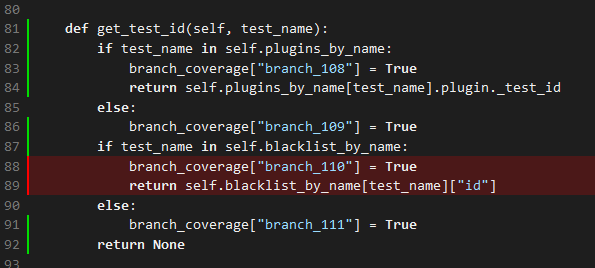
<<https://github.com/Fedor-Baryshnikov/bandit-sep/commit/687b8e59f374b5d5c2bd0c523108db8af98fe8a7#diff-4e949bddd9e1af64c746ad995c0b2b4dae1a008f194b84b7c42244eeff5b3664>>

<Provide the same kind of information provided for Test 1>





<Initially, the function was covered only by 66%. After tests were implemented, the total percentage of coverage improved to 77%, as it is demonstrated at the screenshots. New coverage covered all the branches of the mentioned function. >





### Overall

<Provide a screenshot of the old coverage results by running an existing tool (the same as you already showed above)>

<Provide a screenshot of the new coverage results by running the existing tool using all test modifications made by the group>

## Statement of individual contributions

<Write what each group member did>

Each group member wrote new tests, implemented branch coverage and improved the coverage of the following functions:

Dogaru Eduard-Alexandru:

* test\_code()
* get\_skipped()

Fedor Baryshnikov:

* get\_url()
* from\_dict()

Jia Hao Ji:

* get\_test\_id()
* output\_results()

Egor Vasiliev:

* get\_path\_for\_function()
* visit()