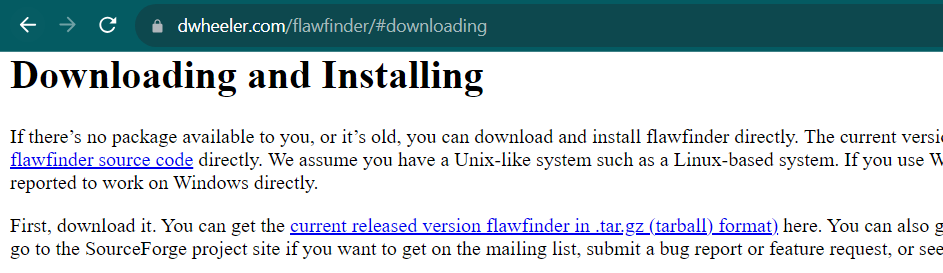
**CYBER SECURITY & DIGITAL FORENSICS**

Practical-1

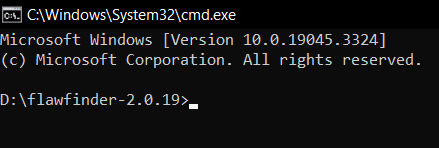
Static code analysis using open-source tool Flawfinder

* **Steps to download required tool- flawfinder**

1. To download the flawfinder visit to <https://dwheeler.com/flawfinder/> and download the current released version of flawfinder in ‘*.tar.gz (tarball)’* format.

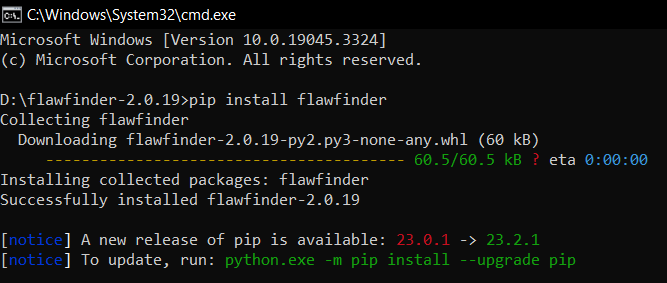


1. Extract the *‘flawfinder-2.0.19.tar’* file and open command prompt within it.



1. Run the following command to install flawfinder:

pip install flawfinder



* **Static code analysis using open-source tool Flawfinder for the following: -**
* **BUFFER OVERFLOW RISK**

**Step 1:**

Create a C file having the following program mentioned below. In this program (the length of secretMessage) is greater than 10 characters, the strcpy function will write beyond the bounds of the buffer array, causing undefined behavior.

#include <stdio.h>

#include <string.h>

void printMessage(const char \*message) {

    char buffer[10];

    strcpy(buffer, message);  // Potential buffer overflow here

}

int main() {

    const char \*secretMessage = "This is a secret message!";

    printMessage(secretMessage);

    printf("Message printed successfully.\n");

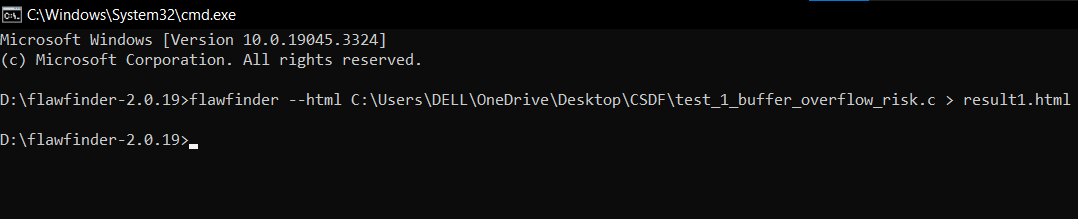
    return 0;

}

**Step 2:**

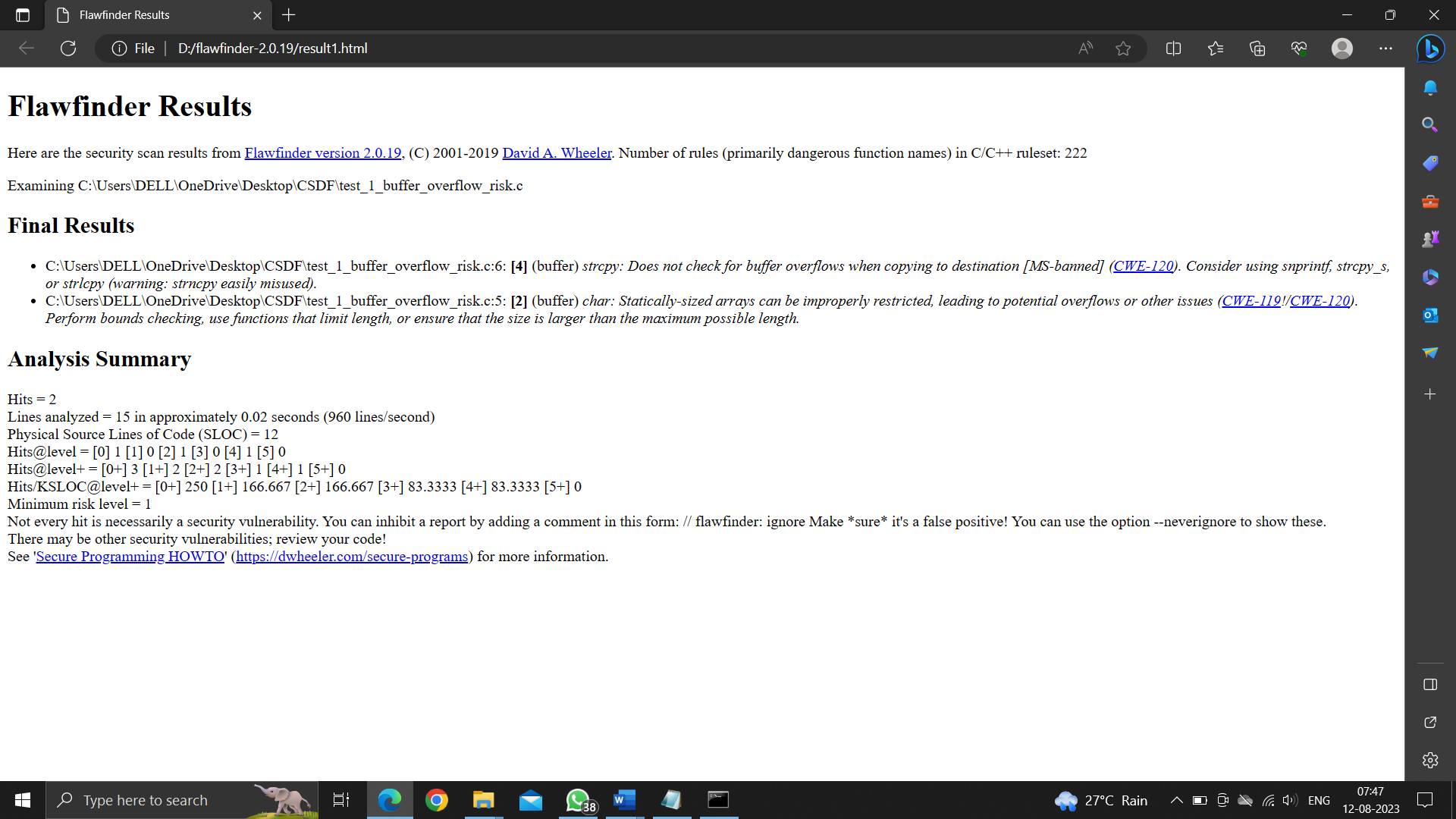
Run the following command to view output in an HTML document using Flawfinder.

flawfinder --html C:\Users\DELL\OneDrive\Desktop\CSDF\test\_1\_buffer\_overflow\_risk.c > result1.html



**OUTPUT**

result1.html



* **FORMAT STRING PROBLEM**

**Step 1**

Create a C file having the following program mentioned below. In this program the line printf('Enter your name:' ); has a syntax error. Single quotes (') are used to denote character literals, while double quotes (") are used for string literals. In this case, you should use double quotes to create a string literal for the printf function. It should be: printf("Enter your name: ");

In the line printf(hello name);, you're trying to use a variable within a string without proper formatting. To include the value of the name variable in the string, you need to use the format specifier %s and provide the variable as an argument. Additionally, you need to enclose the string "hello name" in double quotes to create a string literal. The correct line should be: printf("Hello %s", name);

#include <stdio.h>

int main() {

    char name[20];

    printf('Enter your name:' );// Vulnerability here

    scanf("%s", name);

    printf(hello name);  // Vulnerability here

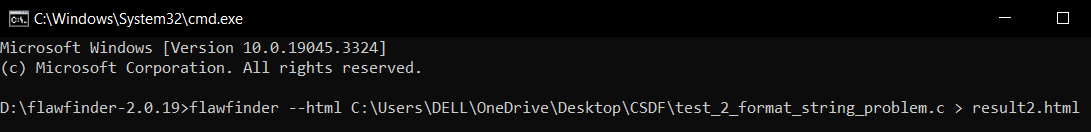
    return 0;

}

**Step 2**

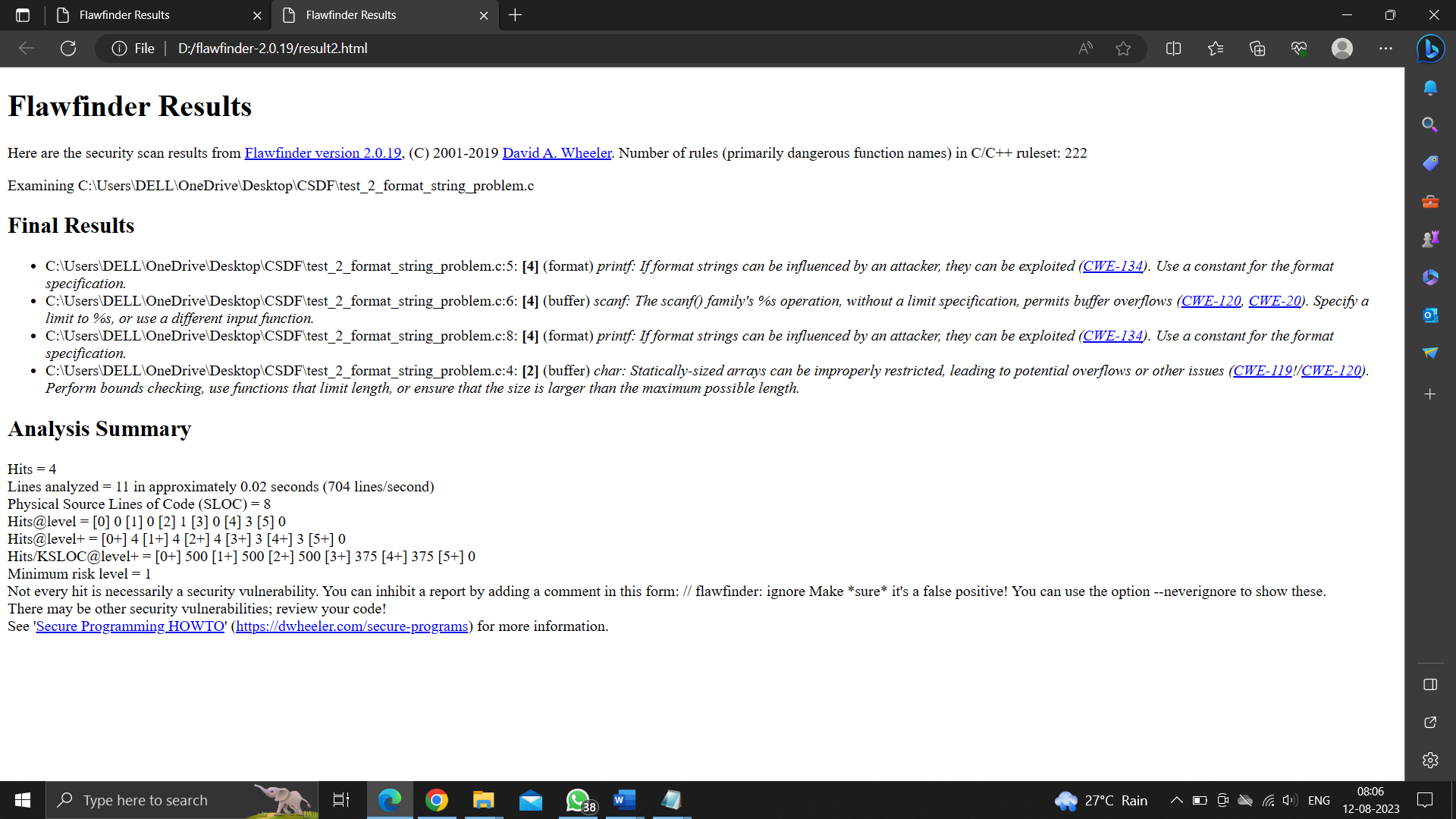
Run the following command to view output in an HTML document using Flawfinder.

flawfinder --html C:\Users\DELL\OneDrive\Desktop\CSDF\ test\_2\_format\_string\_problem.c > result2.html



**OUTPUT**

Result2.html



* **RACE CONDITIONS**

**Step 1**

Create a C file having the following program mentioned below. In this program the race condition occurs between the access function call and the fopen function call. During the short time between these two calls, the file permissions could change by an external process or another user. For instance, if the permissions on the file "/tmp/XYZ" change from writable to non-writable after the access call but before the fopen call, the code will still attempt to open and write to the file, resulting in an error.

include<stdio.h>

int main() {

    char \* fn = "/tmp/XYZ";

    char buffer[60];

    FILE \*fp;

    /\* get user input \*/

    scanf("%50s", buffer );

    if(!access(fn, W\_OK)){

        fp = fopen(fn, "a+");

        fwrite("\n", sizeof(char), 1, fp);

        fwrite(buffer, sizeof(char), strlen(buffer), fp);

        fclose(fp);

    }

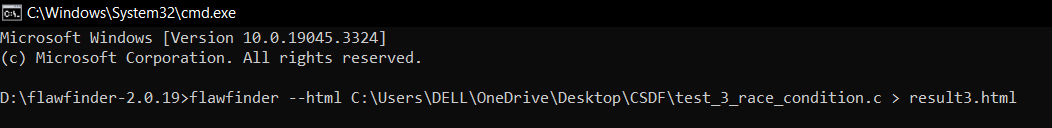
    else printf("No permission \n");

}

**Step 2**

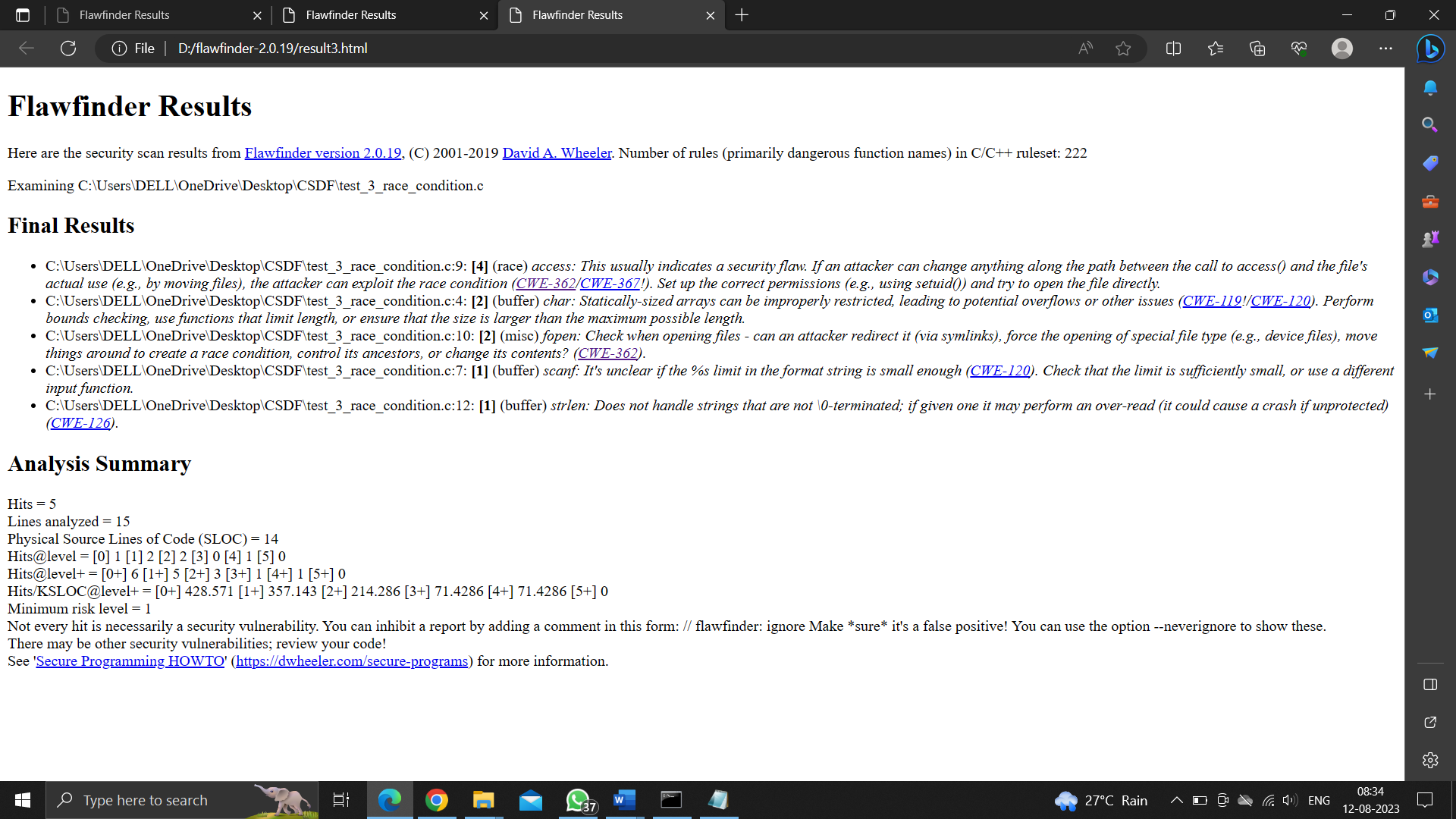
Run the following command to view output in an HTML document using Flawfinder.

flawfinder --html C:\Users\DELL\OneDrive\Desktop\CSDF\ test\_3\_race\_condition.c > result2.html



**OUTPUT**

Result3.html



* **METACHAR**

**Step 1**

Create a C file having the following program mentioned below. In this program the main vulnerability in this code is due to improper handling of user input. If an attacker enters a malicious filename containing metacharacters or shell commands, those commands could be executed when the system function is called. For example, if an attacker enters "myfile.txt; rm -rf /" as the filename, the constructed command would be "cat myfile.txt; rm -rf /", which could lead to serious data loss if the command is executed.

#include <stdio.h>

#include <stdlib.h>

int main() {

    char filename[100];

    printf("Enter a filename: ");

    scanf("%s", filename);

    char command[200];

    sprintf(command, "cat %s", filename);

    printf("Executing command: %s\n", command);

    system(command);

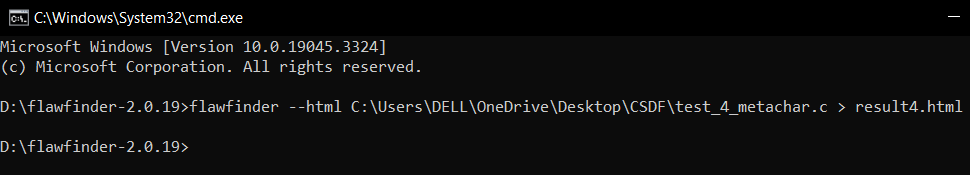
    return 0;

}

**Step 2**

Run the following command to view output in an HTML document using Flawfinder.

flawfinder --html C:\Users\DELL\OneDrive\Desktop\CSDF\ test\_4\_metachar.c > result2.html



**OUTPUT**

Result4.html

