Static code analysis report

Francesco Paolo Di Lorenzo
student ID: 1712990
e-mail: dilorenzo.1712990@studenti.uniroma1.it

Introduction

In this report, a static analysis of a C code fragment is performed using tools such as Splint and Flawfinder.
In the first section there is a high-level description of the tools,which indicates their main strengths and weaknesses.
The second section, shows the output of the respective tools (mainly vulnerabilities and problems) and the resolution of the latter.
The last section presents the correct version of the program obtained by solving the problems reported with the analysis.

1 Static Analysis tools

This section describes main strenghts and weaknesses of Flawfinder and Splint.

1.1 Flawfinder

Flawfinder is a tool for statically scanning C/C++ source code for `possible security weaknesses`. These security weaknesses are called *flaws* or *hits* and are ordered by risk level.
The risk level is shown in square brackets and can take value ranging from 0 (very little risk) to 5(high risk)[1].
Furthermore it is compatible with CWE (Common Weakness Enumeration)[4][3] and may detect many of the most widespread and critical errors drafted in the 2011 CWE/SANS Top 25 list.
Flawfinder is a simple and easy to use tool. This involves some pros and cons.[1]
Unlike programs such as Splint or gcc's warning flags, Flawfinder has no

access to the program control flow, data flow and data type when looking for vulnerabilities. This leads the program to produce false positives or fail to report some vulnerabilities. In his favor, instead, we have that he can analyze programs that cannot be compiled,in a fast and efficient way.

1.2 Splint

Splint is a tool for statically checking C programs for `possible security vulnerabilities and coding mistakes`.[5]
It is very useful for checking type,checking of variable and function assignments, efficiency, unused variables and function identifiers,unreachable code and possible memory leaks.
Splint is a very light static analysis tool, it helps to improve the quality of the code, even if it does not help to eliminate all the security flaws and produces many warnings that can lead to confusion[6].

2 Output description

This section describes the outputs of the respective tools and shows how starting from these outputs, it is possible to improve the code and free it from vulnerabilities and errors that can lead to serious problems.

2.1 Flawfinder output

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit,

vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

2.2 Splint output

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

3 Corrected version of the fragment

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

4 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

References

[1] Flawfinder official documentation,
`https://dwheeler.com/flawfinder/flawfinder.pdf`

[2] The MITRE Corporation. *Evaluation of Static Analysis Tools for Finding Vulnerabilities in Java and C/C++ Source Code*. CWE(Common Weakness Enumeration)[3].

[3] CWE (Common Weakness Enumeration),
`https://cwe.mitre.org/index.html`

[4] Rahma Mahmood, Qusay H. Mahmoud. *2011 CWE/SANS Top 25 Most Dangerous Software Errors*. Department of Electrical, Computer & Software Engineering,University of Ontario Institute of Technology,Oshawa, ON, Canada.

[5] Splint official page,
`https://splint.org/`

[6] Pedro pereira, Ulisses Costa. *Splint the C code static checker*. Formal Methods in Software Engineering, May 28, 2009