PA193 Secure coding principles and practices: Part -1
Mat´uˇs Nemec, Pankaj Agarwal, Anupam Gupta (Team B)

Project Description: The project that we reviewed is "vuurmuur" this is a powerful firewall manager for Linux/iptables. Vuurmuur supports traffic shaping and live monitoring. It has an easy to learn configuration that allows both simple and complex configurations, and can be fully configured through the Ncurses GUI.

Characteristics:

- Mostly written in C
- Line of Code : 66 KLOC

Tool used for analyzing the project: Cppcheck.

The summary of errors/warnings detected by the tools is presented below:-

Sl. No.	Bug Type	Number of occurrence for this error type
1.	The scope of the variable can be reduced.	**162**
2.	Variable is assigned a value that is never used	**10**
3.	Resource Leak	**5**
4.	Array index 'i' is used before limits check	**14**
5.	No input data length check :scanf without field width limits can crash with huge input data.	**9**
6.	Buffer over flow	**10**
7.	Duplicate conditions in 'if' and related 'else if'	**7**
8.	Checking if unsigned variable 'lockpath_len' is less than Zero	**8**
9.	Memory leak	**26**
10.	Possible null pointer dereference	**3**
11.	Suspicious condition (assignment+comparison), it can be clarified with parentheses	**1**
12.	Boolean result is used in bitwise operation. Clarify expression with parentheses	**1**
13.	Redundant code	**1**

The errors classified as per type is as under:-

1. Style type errors – 168
2. Errors – 44
3. Warnings – 10
4. Performance type errors ,Portability ,Information – 0

Analysis of the errors/warnings reveals the following:-

1. Some more serious bugs are related to memory leaks and dereferencing a null pointer, these can potentially lead to exhaustion of memory and segmentation faults respectively, which can arbitrarily close a running application in unstable state.

2. Large number was attributed to style type warnings. Specifically, as found in cppcheck, scope of a lot of variables could be reduced by declaring them close to their use and saving memory.

3. A large number of variables were assigned values but were never used. Also many variables were used without being initialized. These bugs can be easily removed and may not pose a serious threat.

4. Duplicate conditions in 'if' and related 'else if' indicate a possibility of logical errors.

5. Some bugs like assignment within conditional expression can lead to logical error.

6. Large number of warnings related to format string error have been found. It can lead buffer flow

7. Removing of Redundant code can significantly improve a program's quality. While the functionality won't change, it will improve the internal quality – the quality of the source code. This will help in maintenance by decreasing the maintained code size, making it easier to understand the program and preventing bugs from being introduced.

8. Suspicious condition in assignment and comparison can lead to ambiguous result if the precedence of the operator have not been specified clearly and explicitly. use of parentheses will remove the ambiguity in calculating the expression