VulPecker: An Automated Vulnerability Detection System Based on Code Similarity Analysis

Instances of the same vulnerability may exists in multiple software copies for example Flash libraries are used in Adobe Reader and Adobe Air and Adobe Reader may be included in printer driver an if we could find one vulnerability in Adobe may be useful for 3 software!

This paper tries to answer to an important question in software security and implement it:

If we have a given vulnerability and a random source code can we find a way to understand that this vulnerability exists in source code? (vulnerability prevalence problem)

VulPecker is a system for automatic detecting source code to determine that it contains specific vulnerability or not.

This system faces two main challenges:

1-insufficent data set for common vulnerabilities

Solution by author: building a Vulnerability Path Database (VPD) and Vulnerability Code Instance Data base (VICD) which correspond to C/C++ open source products and gather their vulnerabilities.

2-lack of a comprehensive code-similarity algorithm that compatible to all kind of vulnerabilities

Solution by author: we have more than one algorithm! In fact one for each type of vulnerability base on diff hunks features.

There are to approaches in similarity detection:

1-using vulnerability patterns

This method is useless because requires all instances before stating and we number of instances in real word are infinity and it is impossible.

2-code similarity methods

Code-similarity algorithms have three attributes:

1-code fragment level
2-code representation
3-comparision method
Vector comparison
Approximate/exact matching

Part 1 and 2 show the code in abstract model and the third part elect a method for comparing.