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

Software vulnerabilities are the fundamental cause of many attacks. We should notice that same vulnerabilities may exist in multiple software copies also we have code reuse, so it is difficult to patch all vulnerabilities and it is not simple to track code reuse.

One solution for problems that mention above is to automatically identify whether a program contains a given vulnerability or not. VulPecker is a system that can do this work for us.

For detecting the vulnerabilities there are two approaches: using vulnerability patterns, using code similarity.

Pattern based detection is that we need multiple instances of the same vulnerability to recognizing the pattern but using code similarity based detection only needs single instance of vulnerability and VulPecker is based on this approach.

Overview of VulPecker :

VulPecker has two phases: Learning phase and detection phase

Learning phase:

Input:

NVD (National Vulnerability Database),

VPD (Vulnerability Patch Database), VCID (Vulnerability Code Instance Database)

Operation:

1) Code-Similarity algorithm selection

This module finds the most suitable code similarity algorithm by comparing target program and a piece of vulnerability code then find the algorithm with higher similarities that we call it CVE-to-algorithm mapping.
Output: CVE-to-algorithm mapping

2) Vulnerability signature generation
In this module we extract the patched/unpatched diff code from diffs according to the VPD, and also extract unpatched code fragment from the source code of the vulnerable software then preprocess them and generates vulnerability signature with the help of Algorithm selection engine.
Output: vulnerability signature

Detection phase:

Input:

CVE-ID (Vulnerabilities and Exposures number or identifiers)
Target programs, output of Learning Phase.
Operation:
Searching for vulnerability signature in the target program and if exists, report the location of vulnerability.