Vansh Pahuja

CSS lab 8

Aim: To simulate buffer overflow attack using Splint.

Theory:

Buffer overflow is a mistake that exist in some C implementation. These classes of bugs are dangerous as they write past the end of a buffer or array and hence corrupt the process stack.

They often change the return address of a process after a function call to secret memory location where a malicious code is planted

There are 2 main types:
a) Stack based attacks.
b) Heap based attacks.

Heap based attacks are the one which flood the memory space reserved for a program, but the difficulty involved with performing such an attack makes them rare.
Stack based buffer overflows are by far the most common.

→ Splint tool:

1) Splint is a tool for statically checking C programs for security vulnerabilities and programming mistakes.
2) Splint does many of the traditional lint checks including use before definition, type inconsistencies, use before definition, unreachable code, ignored return, execution paths with no return, likely infinite loops and fall through cases.
3) More powerful checks are made possible by addi. info. given in source code annotations.
4) In addition to the checks specifically enabled by annotations, many of the tran lint checks are improved by exploiting additional information.
5) Splint is designed to be flexible & allow programmers to select appropriate points on the effort benefit curve.

→ Problems detected by splint include.

1) Dereferencing a possibly null pointer.
2) Using undefined storage.
3) Type mismatches with greater precision.
4) Violation of information hiding
5) Memory management errors.
6) Modifications & global variable uses that are inconsistent.
7) Problematic control flow.
8) Buffer overflow vulnerabilities.

* Conclusion: Thus, I have successfully simulated buffer overflow attack using splint tool.