

✓✓ = Very effective* ✓ = Effective* ✕ = Not effective N/A = not applicable *if well implemented	Can Catch Buffer Overflows Before Software is Released	Can Stop Stack Smashing Attacks	Can Stop Data Corruption Attacks	Can Prevent Known Attacks	Can Prevent Attacks On Known Vulnerabilities	Can Protect Against Attacks On Unknown Vulnerabilities	Does Not Require Software Modification	Does Not Require Software Recompil ation	Can Help Protect Against Denial-of- Service Attacks
Testing	✓✓	✓✓	✓✓	N/A	N/A	✓✓	✕	✕	✓✓
Code Inspection	✓✓	✓✓	✓✓	N/A	N/A	✓✓	✕	✕	✓✓
Documentation for reused code	✓✓	✓	✓	N/A	N/A	✓	✕	✕	✓
Software Patches	✕	✓	✓	✓✓	✓	✕	✓	✓	✓
Programs that Block Known Attacks	✕	✓	✓	✓✓	✕	✕	✓✓	✓✓	✕
Languages Less Susceptible to Buffer Overflows	✕	✓✓	✓✓	N/A	N/A	✓	✕	✕	✓
Languages Based on C	✕	✓✓	✓✓	N/A	N/A	✓	✕	✕	✓
“Safe” buffers	✕	✓✓	✓✓	N/A	N/A	✓	✕	✕	✓
Safer Library Functions	✕	✓	✓	N/A	N/A	✓	✓	✓	✓
Static Analysis	✓✓	✓	✓	N/A	N/A	✓✓	✕	✕	✓✓
Dynamic Analysis	✓✓	✓✓	✓✓	N/A	N/A	✓✓	✕	✕	✓✓
Automatic Bounds Checking	✕	✓✓	✓✓	N/A	N/A	✓	✓✓	✕	✕
Protection of the Return Pointer on the Stack	✕	✓✓	✕	✓	✓	✓	✓✓	✕	✕
Disabling the Execution of Code Outside the Code Space	✕	✓	✕	✓	✓	✓	✓✓	✓✓	✕
Intrusion Detection	✕	✓	✕	✓	✓	✓	✓✓	✓✓	✕
Generation of an Interrupt	✕	✓✓	✓✓	N/A	N/A	✓	✓✓		✕

This document is part of a larger package of materials on buffer overflow vulnerabilities, defenses, and software practices.

Authored by Jedidiah R. Crandall

©2002, Jedidiah R. Crandall, Susan Gerhart, Jan G. Hogle. <http://nsfsecurity.pr.erau.edu>

Distributed July 2002.

This Document was Funded by the National Science Foundation Federal Cyber Service Scholarship For Service Program: Grant No. 0113627

For more information, go to: <http://nsfsecurity.pr.erau.edu>

or contact Susan Gerhart, gerharts@erau.edu

Embry-Riddle Aeronautical University • Prescott, Arizona • USA

Also available are:

- Demonstrations of how buffer overflows occur (Java applets)
- PowerPoint lecture-style presentations on an introduction to buffer overflows, preventing buffer overflows (for C programmers), and a case study of Code Red
- Checklists and Points to Remember for C Programmers
- An interactive module and quiz set with alternative paths for journalists/analysts and IT managers as well as programmers and testers
- A scavenger hunt on implications of the buffer overflow vulnerability

Please complete a feedback form at <http://nsfsecurity.pr.erau.edu/feedback.html> to tell us how you used this material and to offer suggestions for improvements.