

Analysis Report

McNair Scholar Case Study: Buffer Overflow Security Vulnerability, supplement to CS 332 Case Study

Summary

This is a formative review of the buffer overflow java applets designed for the buffer overflow security vulnerability module. The review supplements a previous evaluation of the applet with an CS 332 audience familiar with computer science concepts. The review in this report evaluates the java applets when presented to a different audience, McNair Scholars, who are generally less familiar with computer science than the CS 332 students.

Results show that the McNair group received the applet presentation as expected for advanced students with little or no prior knowledge of the subject. Most were able to list several key concepts of the buffer overflow problem, but most were not able to describe details of the cause or process of buffer overflows. It would be expected that this audience would have greater understanding and retention of the material if presented with the applets in a user-controlled situation, with supplemental supporting material which can be reviewed at the user's pace.

The McNair audience gave several suggestions for improvement of the applet presentation, many in agreement with the suggestions of the CS 332 group. They asked for supplemental materials, such as a more detailed or simplified introduction, more information about solutions, and a glossary of terms.

Introduction

This report is a supplemental formative review of a student-developed Java applet presented to McNair Scholars by its student author. The applet addressed the problem of buffer overflow security vulnerability.

Analysis was performed as part of an initial evaluation in the development of a Buffer Overflow module for NSF Security grant at Embry-Riddle Aeronautical University.

Roles:

- Professor: Susan Gerhart
- Student Java author: Jed Crandall
- Interactive learning consultant: Jan Hogle

Purpose

The purpose of the evaluation was to examine the effectiveness of the java applet with a potential audience of the buffer overflow module. Results reflect the effectiveness of the applet and the delivery system, not the abilities of the students who assisted in the review process.

This analysis was an formative review to:

- Better understand the effectiveness of the buffer overflow applet when presented to an audience who, in general, do not possess computer science backgrounds. Students chosen to represent

this audience were McNair Scholars, an advanced group of students preparing for graduate study.

- Better understand the preliminary level of knowledge of the buffer overflow problem possessed by an audience such as the McNair Scholars.
- Obtain feedback by the audience represented by the McNair group about the strong and weak points of the applet presentation.

Limitations

This review was intended to obtain student feedback in the early development of an interactive module. Validity of the analysis was not thoroughly examined at this stage due to the timing and nature of this review. The intent was not to obtain a quantitative comparison of pre- and post-treatment results. A more accurate assessment of validity is expected at later development stages.

Reliability of the interview responses appears to be high due to the consistency of feedback given in three separate feedback sessions. Further reliability assessment will be conducted at later development stages.

Methods:

The student author presented the material to the McNair group via an overhead projector, running the applet in a web browser. The author stepped through each example and explained the material to the class.

Following the presentation, students were asked to answer six questions regarding the presentation. Students emailed their responses to the student presenter. Seven students plus the faculty coordinator responded.

Questions

No pre-treatment quiz was given to the McNair group. Following the presentation, the following questions were asked of the McNair students:

1. Had you ever heard of the Buffer Overflow problem before this demo? What did you know about it?
2. Where do think the fault lies for the Buffer Overflow Problem? Who's to blame?
3. Please summarize how a buffer overflow is made to happen and its effects.
4. What recommendations would you make to the U.S. Government to help overcome the Buffer Overflow problem?
5. What were the strong points of the demonstration technique (Jed and his applet)?
6. What were the weak point of the demonstration? Was there anything you totally didn't understand?

Results:

A summary of responses are included in the appendix.

Preliminary knowledge of the buffer overflow problem possessed by the McNair Scholar group was similar in that none knew anything about the problem; and only two had heard the phrase. In contrast, CS 332 students were more likely to have heard of the problem and most knew something about buffer overflow.

When asked where the fault lay for the buffer overflow problem, the McNair students gave several responses. More than half noted insufficient quality control or lack of testing, careless programmers, and consumer ignorance as reasons for the problem. Individuals also noted lack of accountability of software companies to the government, lack of disclosure by companies, and lack of rules for writing standardized code.

The McNair group was able to list several key concepts when asked to describe what a buffer overflow is and how it happens. Most of the students were not able to give a detailed description of the problem but most noted the ideas that a long string or too much code was entered, that code was overwritten and changed, and that the original code was bypassed. More than half understood that the problem caused a computer system to break down in some way or to go awry. Finer details, such as memory allocations and distinguishing data from code were noted by individuals.

As solutions to the problem, about half of the McNair students suggested establishing government testing regulations, better consumer education, and updating old code. Individuals suggested creating standardized code and establishing code writing guidelines, increased research into software security features and security program recruitment.

Like the CS 332 students, the McNair group indicated a high interest in the graphical presentation of the applet. All but two mentioned the visuals and applet examples as the strong points of the presentation. The student presenter, Jed, was also cited as a positive aspect of the presentation.

When asked about the weak point of the presentation, more than half admitted to not clearly understanding what a buffer overflow was, wanted a simpler explanation, or felt the details faded quickly. More than half also wanted to know more information about solutions to the problem and more demos or examples. Individuals requested supplemental materials such as a better introduction to buffer overflow and a glossary of terms.

Recommendations:

Recommendations that follow are based on feedback from both the McNair group and the CS 332 students. The following actions are recommended in development of the buffer overflow module:

- Incorporate the applet examples into an interactive interface, preceded by supplementary materials, including an overview introducing the topic of buffer overflow, links to a glossary, and to articles or web sites for further reading. The glossary and reference sections could be designed to be accessible from within any of the modules or opened as a separate file.
- Expand the descriptive text in the java examples, if possible. An alternative is to design the interface external to the applet to display expanded descriptive text as the applet example proceeds. This may be done in another window, if necessary.

- Embed the applets or open them in a browser window from the authoring interface.
- Applet examples need to be controllable by the user. The user should control how the example “steps through the process”, including starting, stopping, pausing, and reversing the example.
- Prerequisite background should be noted in the opening screen of the module. As noted by the contrasts between the McNair group and the CS students, different audiences possess varying prior knowledge of the buffer overflow problem. This needs to be considered in the development of the supplemental texts, and in development of reference information or links.

Appendix: Summary of student responses

(1) Had you ever heard of the Buffer Overflow problem before this demo?
What did you know about it?

	knew nothing	heard phrase	knew some	knew much
McNair	5	2	1	
CS 332		1	5	3

(2) Where do think the fault lies for the Buffer Overflow Problem? Who's to blame?

Insufficient quality/ testing	Careless programmers	Consumer ignorance	Lack of accountability to govt	Lack of disclosure by company	lack of rules for writing code
~					

(3) Please summarize how a buffer overflow is made to happen and its effects.

Concepts:								
computers can't tell code from data	long string/ too much code entered	"end value" is changed	code is overwritten	code is changed	hacker code is run/ replaces original	memory allocated for code	system goes awry	orig. code is "fooled" or bypassed
~	~							
	~		~					~
					~			

(4) What recommendations would you make to the U.S. Government to help overcome the Buffer Overflow problem?

establish govt testing regulations	consumer education	create universal code	update old code	security program recruitment	research security features	establish code writing guidelines
						~
~						

(5) What were the strong points of the demonstration technique (Jed and his applet)?

[illegible]

