

# Software Engineering Practices as an Approach to Patching Legacy Web Application Vulnerabilities

1<sup>st</sup> Nick Stone  
Xavier University  
Cincinnati Ohio, USA  
Stonen2@Xavier.edu

2<sup>nd</sup> Deep Ramanayake  
Xavier University  
Cincinnati Ohio, USA  
Ramanayaked@Xavier.edu

**Abstract**—Maintaining software can be challenging especially in light of the fact that everyday new vulnerabilities are found throughout the software world. The problem becomes even more complex when a software engineer has the incredible task of patching a vulnerability. The patch that is developed needs to solve the problem while not impacting the functionality of the software product. Additionally, these software patches need to be carefully considered such that the patch does not cause more vulnerabilities as well as the patch ensures that the vulnerability is solved, this meaning that the vulnerability will not come back if a software update is pushed. The API presented aims to bring together software engineering and cybersecurity to evaluate software engineering approaches to patching legacy web application vulnerabilities as well as create a useful and easy to implement API that mitigates common web applications vulnerabilities.

**Index Terms**—API, Legacy Application, Software Patch, Vulnerability

## I. INTRODUCTION

Placement text to start the introduction

## II. VULNERABILITIES IN QUESTION

### A. Javascript Injection

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

### B. Cross Site Scripting

### C. SQL Injection

## III. JAVASCRIPT INJECTION PROTECTION

## IV. CROSS SITE SCRIPTING PROTECTION

## V. SQL INJECTION PROTECTION

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections

Identify applicable funding agency here. If none, delete this.

V-A-?? below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads— $\LaTeX$  will do that for you.

### A. Making the SQL Injection Work For Us

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

### The Importance of Error Handling

## VI. DIFFERENCE IN APPLICATION PARADIGM

### A. Object Oriented

### B. Functional

## VII. WEB APPLICATION ARCHITECTURE

## VIII. THE API

### A. Java

- 1) Javascript Injection:
- 2) Cross Site Scripting:
- 3) SQL Injection:

### B. PHP

- 1) Javascript Injection:
- 2) Cross Site Scripting:
- 3) SQL Injection:

## IX. CAN WE TRUST COOKIES?

## X. EVALUATION

## XI. FUTURE WORK

## ACKNOWLEDGMENT

Thank you to Xavier University for allowing me to perform research on this unusual and uncanny topic.

## REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [?]. Refer simply to the reference number, as in [?]<sup>1</sup>—do not use “Ref. [?]” or “reference [?]” except at the beginning of a sentence: “Reference [?] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [?]. Papers that have been accepted for publication should be cited as “in press” [?]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [?].

## REFERENCES

- [1] M. Shell. (2007) IEEEtran homepage. [Online]. Available: <http://www.michaelshell.org/tex/ieeetran/>

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.