E-Mail Header Injections

An Analysis of the World Wide Web

by

Sai Prashanth Chandramouli

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Dr. Adam Doupe, Chair
Dr. Gail-Joon Ahn
Dr. Ziming Zhao
\memberThree
\memberFour

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

ACKNOWLEDGEMENTS

A project of this size is never easy to complete without the help and support of other people. I would like to take this opportunity to thank some of them.

This thesis would not have been possible without the help and guidance of my thesis advisor, and committee chair - the brilliant Dr. Adam Doupe. This project was his brainchild, and he held my hand through the entire project.

I would like to thank Dr. Gail-Joon Ahn, for being part of the committee, for all his help, and his valuable input on the changes to be made to make the project more impactful.

I would also like to thank Dr. Ziming Zhao, for being a part of my committee, and for the constant motivation.

I would like to thank the members of the SEFCOM, for their support, and would like to especially thank Mike Mabey, for helping set up the infrastructure for this project, and Marthony Taguinod, for helping me document this project.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Just an example to check if citations work Franklin *et al.* (2007) This section will have a brief overview about the project. What the problem is, and What we aim to achieve.

Chapter 2

E-MAIL HEADER INJECTION BACKGROUND

2.1 Problem Background

This section describes the background of the vulnerability.

2.2 History of E-Mail Injection

This section describes the history of the vulnerability.

2.3 Languages Affected

This section describes the popular languages which exhibit this type of vulnerability.

- PHP - Describe which functions/params are affected
- Java - Describe which functions/params are affected
- Python - Describe which functions/params are affected

2.4 Potential Impact

This section describes the impact of the vulnerability, and how wide/far-reaching the effects could be.

Chapter 3

SYSTEM DESIGN

3.1 Our Approach to the Problem

This section will describe the approach we have taken. Will discuss about blackbox testing, and why we chose it.

3.2 System Architecture

This will have a diagram of our architecture, including all 8 components.

3.3 System Components

This will discuss in detail about the components of the system, like the following:

3.3.1 Crawler

Describe the functionality of the Crawler

3.3.2 Form Parser

Describe the functionality of the Form Parser

3.3.3 E-Mail Field Checker

Describe the functionality of the E-Mail Field Checker

3.3.4 Fuzzer

Describe the functionality of the Fuzzer

Non-Malicious Payload Describes what the regular payload is.

Malicious Payload Describes what the malicious payloads are.

3.3.5 E-Mail Analyzer

Describe the functionality of the E-Mail Analyzer

3.3.6 Database

3.4 Test Plan

This section will describe the test plan for the project, and will explain what was tested, and how our system conforms to the requirements.

3.5 Design Issues

This section will describe the issues we might face with the approach that we have chosen, and the design decisions.

3.6 Assumptions

This discusses the assumptions that we have made while building the system, examples include:

1. Crawler is not blocked by the firewalls.
2. The Crawler feed is an ideal representation of the World Wide Web.

Chapter 4

EXPERIMENTAL SETUP

4.1 System Configuration

Will briefly describe the servers used for the experiments.

4.2 Platforms and Software

Will briefly describe the platform, (ie) Ubuntu 14.04, and the softwares that were used for the experiments. (eg) Postfix, Apache, MySQL, etc.

4.3 Languages used

Will very briefly (maybe one paragraph) describe what we used to create the system. (Python 2) Will also describe the limitation of Python, (GIL basically), and point to next section.

4.4 Celery Queues

Will briefly describe how Celery and rabbitMQ help us to overcome the GIL, and do tasks in parallel.

Chapter 5

DATA ANALYSIS AND RESULTS

This section will have tables, images and charts.

5.1 Data

Will display a table/graph with the data, then go on to explain what the fields/-graphs mean.

5.1.1 URLS crawled

5.1.2 Forms collected

5.1.3 Forms with E-Mail Fields

5.1.4 E-Mail received from Forms

5.1.5 Fuzzed Forms

Chapter 6

DISCUSSION

6.1 Lessons Learned

Describes what we learned from this particular project.

6.2 Limitations of the Project

Describes what limitations were present, stuff like:

- CAPTCHAs
- JavaScript Apps
- Blogs powered by WordPress/Drupal
- Mail libraries

6.3 How to prevent this attack

Describes how to prevent this attack, stuff like:

- Use Mail Libraries
- CMS
- Input Validation

Chapter 7

RELATED WORK

This will be a detailed section on the papers that are related to our work, *but* important thing is to show why our work is different from prior work in this area. Also, can/will add references to the blogs and books that describe this attack :)

Chapter 8

CONCLUSION

Conclude with what the results were, whether the vulnerability was widespread or not, and how (if needed) this can be alleviated.

REFERENCES

Franklin, J., A. Perrig, V. Paxson and S. Savage, "An inquiry into the nature and causes of the wealth of internet miscreants.", in "ACM conference on Computer and communications security", pp. 375–388 (2007).

Pietraszek, T. and C. V. Berghe, "Defending against injection attacks through context-sensitive string evaluation", in "Recent Advances in Intrusion Detection", pp. 124–145 (Springer, 2005).

Zanero, S., L. Carettoni and M. Zanchetta, "Automatic detection of web application security flaws", Black Hat Briefings (2005).

APPENDIX A

CODE SNIPPETS