Bug Classification and Vulnerability Dataset

* Mission Statement

To help security researchers communicate clearly about security vulnerability classifications by providing a categorized bug corpus for research purposes

* Abstract

Currently, the largest and most extensive datasets of bugs cover an incredibly broad area. They are leaving out key detail about specific vulnerabilities and are generally lacking in both accessibility and organization. The dataset we create will focus on a specific program, retrieve and organize relevant bugs, and classify the information in a way that is readable and accessible. Bugs were pulled from Mozilla Firefox’s online BugTracker using a Python script with a Selenium plugin. After being transferred to an excel sheet, bugs were individually classified based on our predetermined vehicle and effect classifications. We later collaborated over the data and resolved disputed classifications. Our original categories for classifications didn’t encompass all the bugs, and we ended up adding a vehicle category specifically for UI Issues. This category totaled roughly a third of all vehicle classifications, and by extension led to our two new UI effects, UI Hardening and UI Misrepresentation. Moving forward, there seems a disconnect between what developers, researchers, and end-users all believe constitutes a security bug.

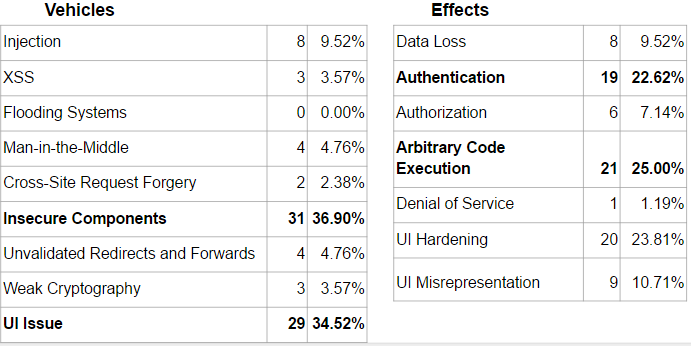
* Introduction

As it currently stands, there is no cohesive list of security vulnerabilities available for any specific program. The CVE (Common Vulnerabilities and Exposures) and the NVD (National Vulnerability Database) both don’t satisfy the needs of researchers, both casting too wide a net to contain specific details about a program’s security issues. Most companies do keep a log of discovered and fixed bugs, but that fails to provide categorical details that can be used to graph vulnerability trends or quickly solve similar bugs in the future. Overall, current datasets of bugs are disorganized and lack the proper details and definitions needed to satisfy the needs of researchers. The goal of this new research is to create a dataset of all the bugs of a specific program that is organized, accessible and highly detailed. Each bug will be classified according to its method of exploit and its effect on the system as a whole. After the dataset is finished, it will be saved to to the Promise Repository for all future researchers to use.

* Research Methodology

First, the data from Mozilla Firefox’s online BugTracker needs to be parsed into a CSV file. Using Python scripts with a Selenium plugin, the bug ID, title, and associated file names were pulled. The CSV was then transferred into an excel sheet, which was later hosted on Google Drive so Chris and I could collaborate. We each individually classified all the bugs, then later came together to resolve disagreements. The final classifications are saved in the excel sheet. The groupings used were derived from the OWASP Top 10, and further iterated and discussed by Chris and myself. We ultimately used 9 different vehicles and 7 different effects, categorizing each bug independently.

* Results



After excluding irrelevant bugs and bugs without code solutions, we were left with 91 bugs in our final taxonomy. As we did our classification, we ran into an issue with a large amount of bugs that didn’t fall into any of our set categories. To rectify this, we decided to add a separate category for UI issues, which ended up as the classification for a third of our total bugs. We found that we agreed on classifications about 73.21% of the time.

* Limitations

One bug in the dataset could have easily fit into many categories, especially for the effect. We opted to use the first-order typing to deal with bugs with compound vehicles or effects, although a classification system that allowed for multiple typings would be optimal. Also, I hardly consider myself a cybersecurity expert, yet I was one half of the collaborators working on the project. All my collaboration was done with Chris, who taught me everything I know about cybersecurity. Since he was my teacher, any of his biases are likely now my own, and also probably appear in the dataset.

* Future Work

The 91 bugs that have been classified are only a small subset of the total Firefox BugTracker’s data. As the taxonomy continues to grow, new bugs will need to be categorized. Although right now this requires manual classification, eventually this process can be automated by parsing for keywords in bug comment sections. The scripts will have to be altered to make this happen, which would be a small project in and of itself.