**CSE 591: Security and Vulnerability Analysis**

**Project Report**

**Group 11:403:Forbidden- Amol Deshpande, Avaneesh Desai, Ankit Doshi, Pooja Shastry**

**How much of your project was your team able to accomplish?**

Our team was able to accomplish 70% of the project goals that were proposed. We extended an open source tool called Grabber and made it work to detect SQL injections, XSS Attacks and file inclusion attack. We improved the tool by fixing bugs in the parsers, enhanced the spider to crawl and correctly extract the URLs. We also implemented cookie support that was not previously present that allows to scan pages that needs authentication. We developed an interactive HTML report generation module for the tool to report the detected vulnerabilities in an intuitive manner.

We also developed an automated submission script to submit flags every five minutes. It was a python script to fetch flags from all the vulnerable spots, make GET requests to vulnerable page and return the response. This script saved us a lot of time thereby allowing us to concentrate on other services.

**How did your project help your team in the FinalCTF?**

The project was used to scan vulnerabilities in the FinalCTF. Automated submission script allowed us to submit flags every 5 minutes thus scoring more points.

**What would you change about your project to improve its use in the FinalCTF?**

We would like to add automatic login support to the tool. As of now, we need to manually login and include cookies in the command line execution of our tool. Automatic login support would eliminate the need of cookies by creating user by detecting a register user page and login with same user that will allow us to scan the pages once user is logged in. We would also like to extend our tool to perform extended analysis of the JavaScript code by plugging in a JavaScript engine like spidermonkey.

**What are your group's plans for the project after class?**

Our tool is an already an open-source tool. We will notify the original developer about the changes and we will continue to improve it. Our goal will be to add the functionalities we had thought about while deciding the project idea.

**Anyone on your team deserve special recognition for going above and beyond the call of duty, either in the project or the FinalCTF?**

We all tried hard to make this tool run and we all were equally excited about the FinalCTF.

**Anyone on your team not pull their weight or do their fair share of work?**

We all collaborated perfectly and made sure that no one falls short or is left out while discussing about this project.

**Thoughts and/or comments on the project, FinalCTF and class as a whole?**

SVA is one of the best classes we have taken so far at ASU. It taught us all the loopholes an application might possibly have. We enjoyed the assignments as we were learning a lot while not worrying for the score. FinalCTF is one of the most innovative ways of testing our abilities. We really wish we could have such hackathons in most of the courses. The project was a challenge and we are glad that we were able to sail through most of our plan.

**Appendix**

**Vulnerability in test-service**

**How we found it:**

test-service returned the hidden flag when the following GET Request was sent to the service:

http://10.41.X.2:9000/test-service/test=../../flag

**Description of the Vulnerability:**

This vulnerability was a file-inclusion attack. The php code directly took a GET parameter without performing any sanitization routines. This enabled us to pass a parameter ‘../../flag’ which performed a path traversal in order to retrieve the flag.

**Working exploit of the Vulnerability:**

We used an automated script which we ran every five minutes in order to retrieve changing flags of all the teams. Our code was written in Python and it sent a GET request to the service.

**Patch for the Vulnerability**

We fixed the code which returned the $test variable only if the user entered a valid GET parameter. The patched code is as given below:

***#!/usr/bin/php-cgi***

***<?php***

***if (!isset($\_GET['test']))***

***{***

***?>***

***<h1>Hello Hacker</h1>***

***<p>Welcome to the test</p>***

***<p>Try a <a href="?test=yes">test</a></p>***

***<?php***

***}***

***else if($\_GET['test'] == "yes"){***

***$test = $\_GET['test'];***

***include($test);***

***}***

***?>***