Web Penetration Testing

Project Semester 1

Stefan Zetko

2019

ABSTRACT

I am writing this report to talk about my work done on Web penetration testing for project semester one. To do so, I researched several different web vulnerability and automated testing tools which can be used to test web applications for bugs and vulnerabilities. The main tools I used were ZAP and Selenium. I wrote this report to talk about my experience in project one.

The Background

The internet is defined as the worldwide interconnection of individual networks operated by government, businesses and private parties and has become the universal source of information for millions of people. But with all this information means huge security risk. Organizations continuously strive to protect important information of customers, such as credit card numbers, contact information and many others. Over the years the biggest web application vulnerabilities continue to stay on top with the likes of SQL injection, cross-site scripting and broken authentication. Attackers can use these vulnerabilities against web applications to maliciously use this important information.

The purpose of my project is to efficiently scan web applications manually and automatically using ZAP and Selenium and to have knowledge on how to further fix and avoid any unwanted web application vulnerabilities. The automated scanning process is usually faster, more accurate and saves resources compared to manual testing. So the automated scanning process was more important for me to understand and implement.

The scope of my project was mainly limited to test web applications with the sole purpose of running vulnerability scans against. The reason for this was because it was safer to run active scans against test web applications rather than remote organization applications. During the semester I made a mistake of running an active scan against the Otago Polytechnics website op.ac.nz. I didn’t realise at the time but was told by my lecturer Faisal that this could be dangerous as running an active scan can be noisy, and it was particularly bad timing as the website already had security threats made against it in recent times.

Personal Experience

Web penetration testing involved using a program called ZAP and Selenium. ZAP (Zed Attack Proxy) is one of the world’s most popular free security tools and is actively maintained by hundreds of international volunteers. It can help you automatically find security vulnerabilities in your web application while you are developing and testing your applications. With ZAP there are many different options to use. One example is a quick to use passive scan of a web application by entering a URL into the ZAP interface and receiving a detailed outline of any possible vulnerabilities, with this information you can further protect your application by fixing these bugs. I used ZAP as a proxy for when I used a browser, which basically means that ZAP was the middle man between the web application and my machine. ZAP would automatically run a passive scan against any web application I visited and provided any feedback on potential risks of applications as I surfed the web. However, this was quite limited as when doing so I had to change the browsers network settings and remove the default proxy used. This meant that visiting most websites resulted in a security risk and at times I had to implement a security exception to gain access but mostly even this would not work. I was then given the task to start working with a different tool called Selenium. Basically Selenium is a tool used for automating the testing process of local web applications and testing for any bugs in the code. This would save time for developers as when they were developing they could run a simple script through selenium which would run through the application and outline any bugs throughout the code. Running scripts in headless mode proved to be the most productive as this saved processing power. Headless mode meant that the scripts would be run in the background and the user would not actually see the browsers load and run the application. I ran a small test which proved the efficiency of running scripts in Headless mode vs non-headless mode. When I first ran the script in non-headless mode it took 10.7seconds while in headless mode it took 9.1seconds. The difference may not sound like much but organizations could be running these scripts thousands of times a day, and in a big organization every second counts. Once setting up Selenium was completed and simple example scripts on browsers were run I decided I wanted to integrate Selenium with ZAP so that vulnerability tests could be run at the same time as selenium. This proved to be a tough task as there was limited documentation online to help with this process. I eventually came across some documentation with enough information to provide me with a framework which set up the integrating process of the two tools. However, after spending many of hours working on this I could not get ZAP to run as a proxy for the Selenium scripts. I suspect there was a problem with the setting up process as there were many finicky details that I had to cover. My plan for next semester is to complete this task and to create and run scripts against Web Project students’ applications.

Conclusion

To conclude, the web penetration testing process is an incredibly important process which is only going to become bigger in the future. Using tools like ZAP and Selenium can speed up the testing and securing process of applications extensively resulting in time saved, money saved and most importantly user information kept safe. I would highly recommend ZAP and Selenium as these tools are both free to use and open source which means they can be redistributed or modified if needed. I have learnt a huge amount about web application security as a whole throughout my first semester of project. I look forward to carrying on my project in the second semester and to further broaden my knowledge on the subject. I have goals next semester to implement automated web scanning tests on the Web project students’ applications and provide accurate feedback.