Student name: Adam Logan Student Number: 40293585

CSC3063 – Secure Software Development

Lab 2 Feedback (30%)

First some generic feedback:

Again, the quality of the submissions was very good. It is obvious that most students took care to apply the feedback from the previous assessment into account when writing the report for this lab. Many students didn't put enough emphasis on reporting in the executive summary the impact that a SQLi attack may have on a system, how it can lead to a total loss of control not just of the database, but of the entire OS, to a potential attacker.

One major problem identified in many the submissions is a confusion between "prevention" and "mitigation". Prevention techniques are used to stop the success of an attack, for example appropriate input and output sanitation of user entered values. Mitigation strategies aim at minimizing the impact a successful attack would have, for example using a password hash in combination with a salt. It is clear from the discussion in the reports that many students consider mitigation equivalent to prevention, and this is not correct. Hopefully the above examples serve to clarify the difference. When classifying a technique as mitigation/prevention ask yourself: will this completely stop the vulnerability from being exploited? If the answer is yes, then you are talking about prevention. Will this just make it more difficult for the exploit to succeed? Then you are talking about mitigation.

As students were working with several different boxes, I focused my feedback on the actual quality of the report, rather than specific feedback for each student this time. Generally speaking, most students focused on a handful of machines, with almost every machine related to SQL in some way so the above comments are relevant. Other comments are missed opportunities to discuss how frameworks, such as Django or Laravel in the back end (or for example React in the front end in the case of XSS), can be used to reduce the attack surface of the application. These frameworks apply an abstraction layer to the RDBS, by using an ORM or Object Relational Mapper. This technique allows the programmer to use object-oriented techniques in the database interaction, abstracting tables to models and using getters and setters to manipulate the data in the RDBS without using SQL directly, apply semantic validation to the data and thus minimizing the chances of a SQLi (or a XSS attack in the case of React).

Finally, referencing has improved significantly from the previous report, and more resources have been used to back up arguments and discussions. Well done.

In the next page you will find the feedback specific for your work:

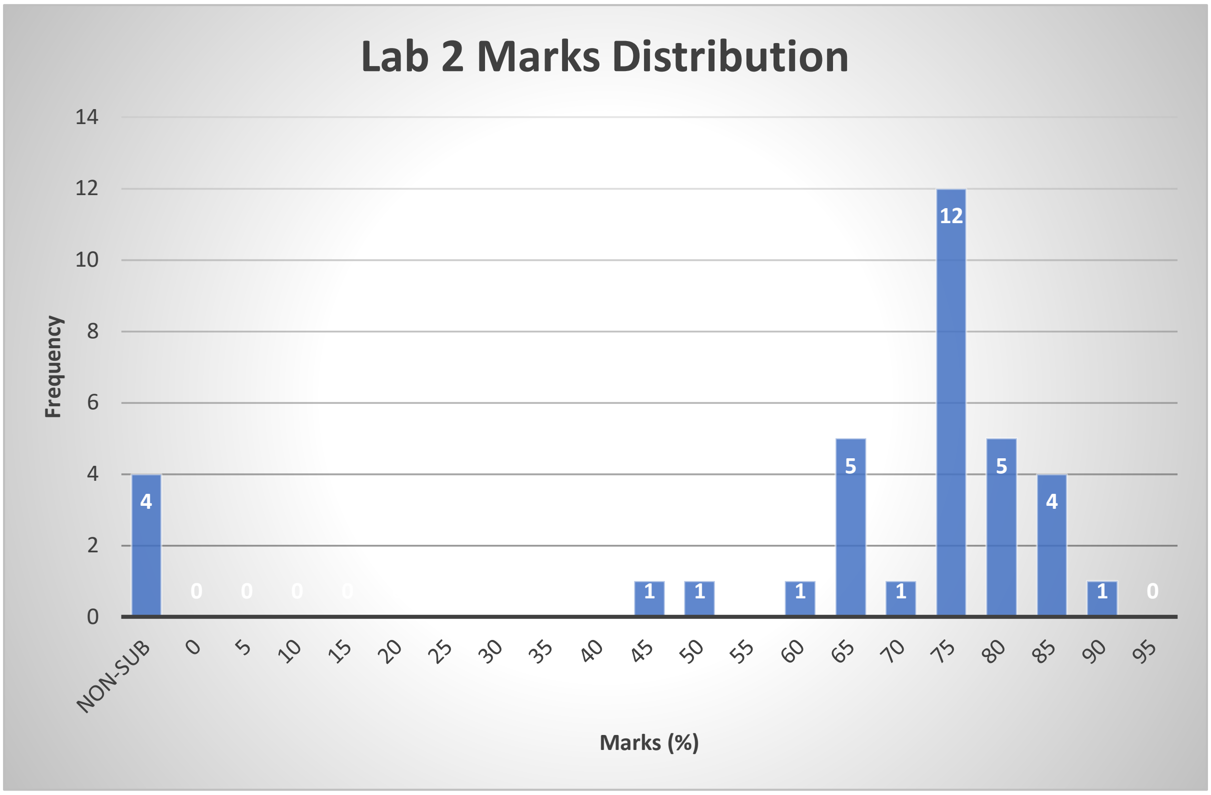
Adam Logan

Feedback	*Assessment Mark*
Followed format requested. Adequate Executive Summary. Good Discussion with some evidence of original learning resources. Clear explanation of the methodology used, shows some confidence in the approach supported by relevant evidence and resources. Good discussion of the vulnerabilities found. Good actions taken, showing satisfactory ability to address the main issues. Excellent approach to mitigation of the vulnerability showing in-depth knowledge and understanding of the wider issues and their relevance. A good level of independence of thought and critical judgement and a level of critical analysis. Excellent referencing using a consistent strategy for reporting and citing sources. The box selected is vulnerable to a SQL injection attack, not necessarily a blind one. You were able to gain access just by commenting out part of the query with the introduction of special characters. Also, you haven't shown your "code rewrite" as you do not have the source code for the application. You have shown some examples of how to perform some input sanitation. Just be mindful of what you report in the future. You missed an opportunity to discuss using salts as part of the mitigation strategy to protect hashed data. You could have also discussed error handling, the role it plays on being able to perform these tests and in mitigating the attacks. You could have also discussed using frameworks as part of the development, or introducing security requirements early in the development process. The overall level of work in this task is good - well done.	**68%**

For reference, here's the mark distribution for this assignment:

You were ranked
#25 of 35

Student name: Adam Logan   Student Number: 40293585



The marks represent the lower limit of the range for example 90 represents marks >90 but <=95.