Threat Analysis: Course Feedback System

Jaan Tollander de Balsch – 452056

October 16, 2018

This summary analyses the security threats to a course feedback system. The feedback is given using an online survey. A unique feedback link is sent through email to all students. The collected data is anonymized. University's goal is to have *a well functioning course-feedback system that will benefit everyone*. This dictates the priorities of the *security goals* for the system.

The main priority should be on the **availability** of the system, i.e, the system shouldn't break or become unavailable for any reason. The system could become unavailable due to server outage or denial of service attack. The feedback fields are potential vulnerabilities to the system. If the inputs are not handled correctly, a malicious user could try using code injection and potentially crash the server, cause data breaches or losses. Inputs should also handle Unicode characters correctly. Serverside checks should also make sure that users cannot input wrong inputs, e.g. too long inputs or wrong values.

Integrity is also important for a well functioning system. Unauthorized users shouldn't be able to send feedback and authorized users should only be able to send feedback once. Otherwise spamming could make the system useless by burying the real feedback under the fakes.

Since the data of the feedback is anonymized, **confidentiality** is also a security goal. The people reading the feedback shouldn't easily be able to figure out the identity of the person giving the feedback. The only hard link back to the person giving the feedback is that it has the effect on their grade. However, the feedback data is not very sensitive, and therefore there isn't much incentive to try to deanonymize the feedback.

In conclusion, the damages of a potential attack could be a loss of time, loss of data, loss of user confidence and engineering costs. Potential attackers would be curious users, script kiddies or hackers since there is no material gain to motivate an attack. Poor software engineering and not testing the system enough could be the biggest threat to the system. These threats can be minimized by testing the system thoroughly, having regular data backups and hosting the system on a reliable server.