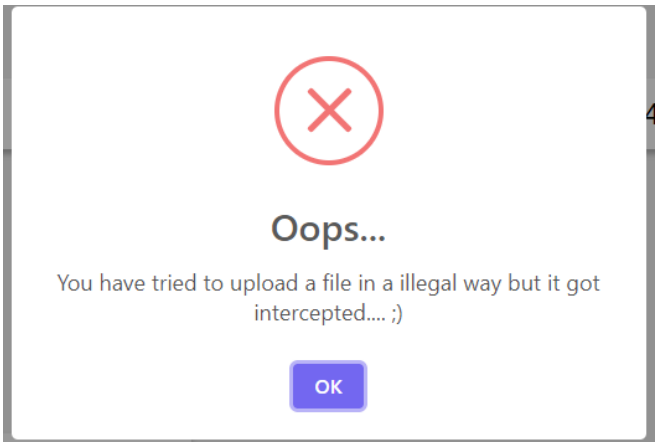
Security report

Security breach	Covered?
Protection against malicious file uploads	Yes
Protection against Man-in-the-middle attacks	Yes
Protection against Link Injection Protection	Yes
Protection against Attribute autocomplete	Yes
Protection against SQL injection	Yes

Protection against malicious file uploads

Problem: Some web applications allow you to upload files. This means that we could upload a file with a malicious script. That server where you have uploaded the file may be compromised. Normally, and this is recommended, the servers usually have different characteristics to validate the uploaded files. For example, detect possible extensions that may be a danger or the type of content.

Solution: To tackle this issue, we first validate each MT940 file before uploading it to our database. In addition, to be able to upload a file one must be logged in. If someone with valid credentials tries to upload a malicious file, they will be prompted with the following:



This protects against potential invalid files that could break our system and against malicious files.

Protection against Man-in-the-middle attacks

Problem: A potential intruder could access website traffic that is transferred in plain text. This is a major privacy issue for users. Basically, an attacker can intercept requests to send information. If we send a message, upload a file or any kind of request, it can be intercepted. This occurs when the traffic is going over HTTP. This way it is not encrypted.

Solution: Our software is made to run inside the bank intranet. We left the HTTPS configuration out as our product will not be deployed on the World Wide Web. However, we use JWT tokens to securely transmit information between the server and user.

Protection against Link Injection Protection

Problem: Another major problem affecting applications and websites is link injection. This could endanger our security and privacy, since we could be accessing a link controlled by hackers. How does this happen? It basically means that cyber criminals inject fraudulent links on that site. In this way, when the victim enters and accesses that link, he is not really entering a website or section that is legitimate but is directly accessing a page or server that is controlled by the attackers.

Solution: As specified earlier, our software is designed to be used by authorized personnel only. If there is an inside job (i.e., an employee has a malicious attempt to delete the files) there is nothing we can do, it is like cutting the wires in the server room, not our issue. All queries use prepared statements.

Protection against Attribute autocomplete

Problem: It is also another type of attack to abuse the autocomplete attribute that is usually set to off mode. The point here is that a potential attacker could have it activated, and this would allow the browser to cache information entered by the user. What could happen to this? A possible attacker would have access to the username and password entered in the browser's cache.

Solution: We have set our autocomplete off.

SQL injection protection

Problem: SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

Solution: All our queries use prepared statements.