

**Finding Name:** Use of hardcoded credentials.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Gaurish Bhatia | SCR | SCR team member | Ontrack | Jaspriya Kaur and payas paul. |  |

|  |
| --- |
| **Was this Finding Successful?** |
| Yes |

**Finding Description**

Hardcoded credentials vulnerability occurs when login credentials are embedded directly into software's source code or configuration files, instead of being securely stored. Attackers exploit this weakness to gain unauthorized access to systems or data, posing significant security risks due to the ease of exploitation and difficulty in updating credentials.

**Risk Rating**  
Impact: Minor  
Likelihood: Unlikely

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

If the code containing the hardcoded email address is leaked or accessed by unauthorized individuals, it could be misused for malicious purposes. This could involve unauthorized account access, spam campaigns, or impersonation attempts.

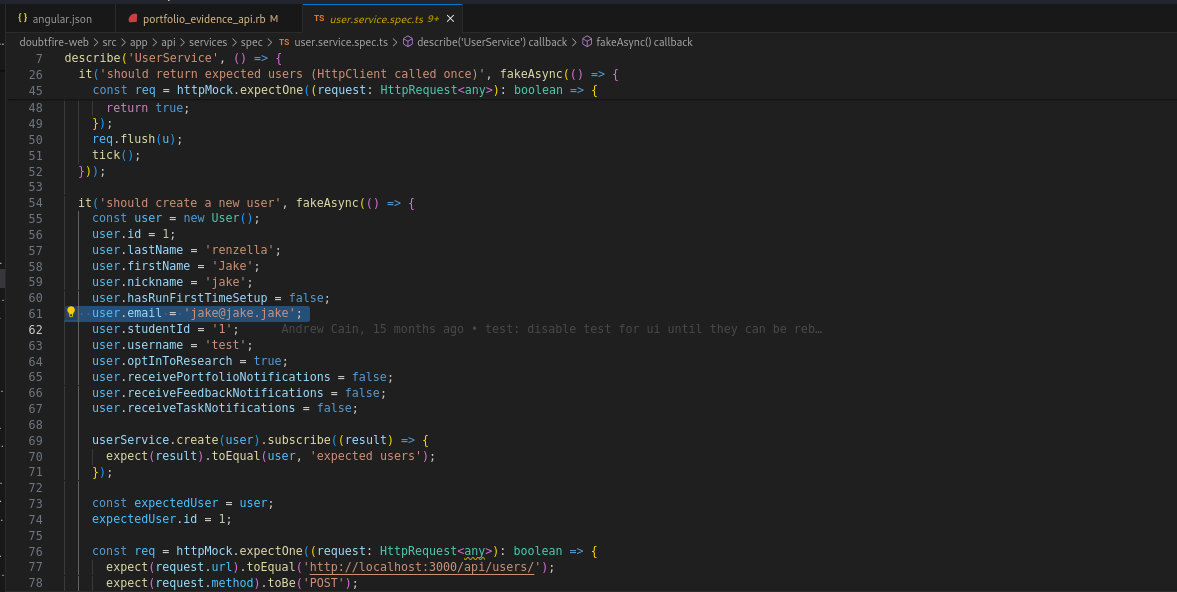
**Affected Assets**

The doubtfire-web application's user service unit tests.

The integrity of the user accounts within the application (if the email is used for a real account).

**Evidence**

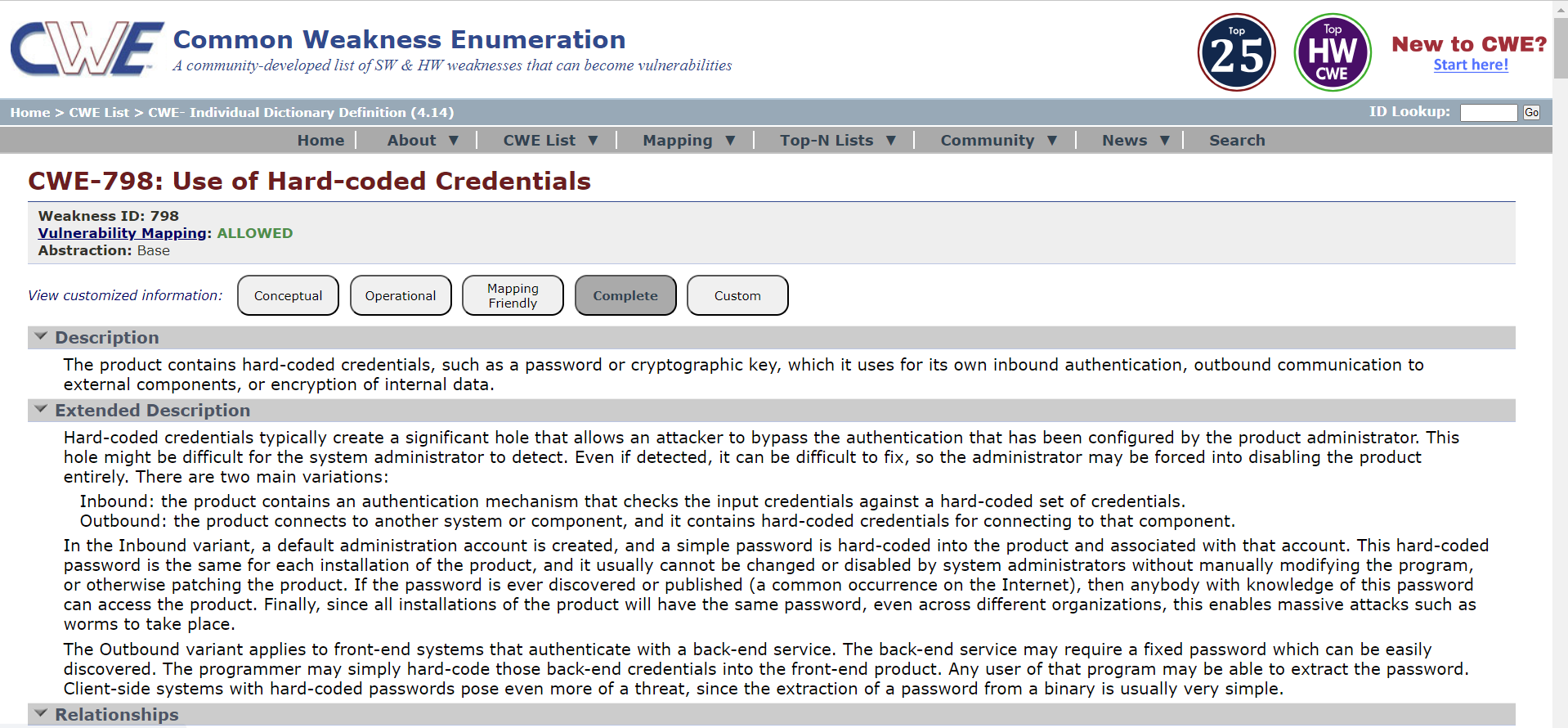
The following screenshot shows the line of the vulnerable code :

****

**A screenshot of a computer

Description automatically generated**

As it could be seen on line 61 and 113, the email ID is hard coded, which could lead to a possible attack by using the same credentials. The path for the file is **doubtfire-web/src/app/api/services/spec/user.service.spec.ts.**



**Remediation Advice**

The vulnerability can be mitigated by following:

Remove the hardcoded email address ('jake@jake.jake') from both occurrences in doubtfire-web/src/app/api/services/spec/user.service.spec.ts (lines 61 and 113).

Implement a secure credential management solution like environment variables or a secrets manager to store test data credentials securely.

**References**

[**https://cwe.mitre.org/data/definitions/798.html**](https://cwe.mitre.org/data/definitions/798.html)

**Contact Details**

Gaurish Bhatia

S222187151@deakin.edu.au

**Pentest Leader Feedback.**

The lead will provide feedback to enact on.