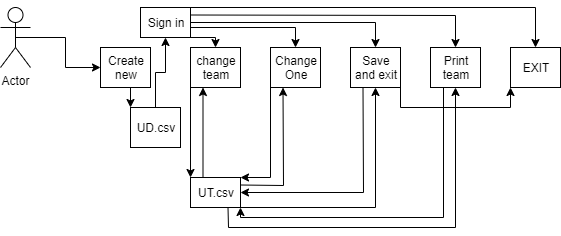
Purpose

This document is designed to show the data flow of the application I have created called pokemon Team Picker. The application is a simple console application that allows a user to log in or create an account and manage a team. Both teams and user details are stored in separate .csv files that are stored in the base directory of the application inside the debug folder.

The application allows a user to have a default team created for them and allows them to change the full team or a single team member, they then have the choices to print the team to screen or to save and exit or exit.

Scope

Architecture & Dataflow



The above diagram shows the flow of the application for a standard user.

* The user creates an account and is then prompted to sign in.
* The user then receives 5 options.
* Change team changes all the values but only in the session, saving must be done through option 3.
* Change one changes one of the values in team but only for the session, saving must be done using option three.
* All information is stored on the user’s device for all accounts created on the device. The methods are built so that id .csv files are hosted online a simple change of directory in the application will prevent users simply looking at the DU and UT files
* Option three saves any changes to the users UT files and exits the application.
* Print team changes some session variables for the user to see the new changes to the team before performing any other actions.
* EXIT closes the application without saving.

Dataflow for each section

A close up of text on a black background

Description automatically generated

Threat Model

**Trust boundaries**

Application user

At present the user of the application has full access to the encrypted files with enough searching. Data can be changed by the user though the user will not know whose date they are changing.

Application user

The user can gain access to the \_PK variable stored in the application that is the private key password that is needed to pass into the encryption and decryption methods to access user data.

Threat Agents

Users

A user may wish to prevent a user from accessing accounts by swapping the fields around in the .csv files prevent a user from login in, the application has very little purpose to other users so this is unlikely and there is potential of preventing there own login due to not knowing there details when encrypted.

External interference

If a malicious file designed to attack .csv files infects the computer, the storage of all user data can be damaged.  
general threats that can affect the OS of the machine the app is stored on.

Assumptions

* The device that has the application stored on it has a competent anti-virus software installed and runs frequent checks.
* The users of the application have not got malicious intent towards other users data.
* The data of each user has been hidden, users do not wish to find the storage place of the data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Category | Trust boundary | Threat Description | Action | Action Description |
| 1 | Spoofing | User on has their own login details | Users logging in with another users’ details | Accept | A user could acquire the login details of another user through means of asking, threatening, blackmail or torture |
| 2 | spoofing | User keeps own details safe | User logs into another user account | Mitigate | User will have to be asked to change password through other means such as email to be implemented |
| 3 | Tampering | Access the UD.csv file and change data around or change user role | The user changes the role that is give then from User to Admin | Mitigate | The Files have been set to hidden so most users will not see the file when searching |
| 4 | Tampering | Access the UD.csv file and change data around or change user details | The user changes around the values in the csv file preventing users from signing in | Mitigate | The Files have been set to hidden but even if found the file is encrypted so no user can tell whose details is being edited. |
| 5 | Tampering | User has access to both csv files and uses them | The user can compare details of the from the two files to identify two similar encrypted username and edit the details of that user as the team is stored as a series of number unencrypted | Accept | With the files hidden from average users this is unlikely and with this effecting a user very little there is no reason for a user to do such. |
| 6 | Repudiation | User knows the made changes | A user who makes changes to their team has no log of the changes happening | Accept | The user has no log of the changes made so they can deny making changes. With this application I see not why a user would do such. |
| 7 | Information disclosure | Users keep own details safe | A user discloses their details to another user | Accept | The application dose not hold vital information to the users though with no change of password option disclosing details mean access till the alt user forgets the details |
| 8 | DOS | User has no need to open 2 apps | If the application is opened twice and ran to the csv reader an exception is thrown in regards to multiples access the csv file. | Accept | There is no reason for a ser to want to open 2 applications. |
| 9 | Elevated Privilege | User has no need for more control | The user changes the User line in the UD.csv to Admin | Accept | No proper information or tools is available to an admin at present. |